Model Order Reduction for Hyperbolic Conservation Laws

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde

(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Roxana-Geanina Curtici

aus

Rumänien

Promotionskommission

Prof. Dr. Rémi Abgrall (Leitung der Dissertation)

Prof. Dr. Sidhartha Mishra

Prof. Dr. Stefan A. Sauter

Zürich, 2018

Abstract

Reduced basis (RB) methods can alleviate the cost of repeated simulations with limited computational resources and are directly based on the underlying high-dimensional model that results from standard finite element, finite volume or finite differences formulation. These methods restrict the solution to be contained in a subspace of the underlying high-dimensional space, this subspace being determined by an optimal reduced basis in a training phase. Thus, a large number of degrees of freedom (say millions) are represented by only a few number of coefficients, which in combination with the reduced basis vectors will lead to important computational savings.

The case of systems of hyperbolic conservation laws is a special, challenging one because in this context, moving waves and discontinuities such as shocks will depend on different parameter settings and they will evolve in time. Hence, accurate surrogates have to be developed, in order to be able to capture the evolution of the discontinuous solutions, which implicitly involve nonlinearities.

The objective of this thesis is to propose innovative ideas for advancing the state of the art of *model order reduction* (MOR) for first-order hyperbolic conservation law problems which are characterized by sharp gradient and shocks. Firstly, we prove that MOR using L^1-norm minimization of the residual leads to a more accurate reduced solution, L^1-norm being a natural norm for evolution problems which involves discontinuities. In order to reduce the Kolmogorov N-width of the solution manifold, we consider a dictionary approach based on no compression when generating the reduced basis functions. In order to emphasize the accuracy of the method we are also providing robust error estimators for the scalar problems in the case of monotone schemes and we illustrate the behavior of L^1-norm minimization based on a dictionary approach on linear and nonlinear problems, both in one and two dimensional case.

Secondly, the parameter dependency of the system of hyperbolic conservation laws means that for different parameter inputs, the position and the shape of the shock is changing. Because of this behavior, we might encounter discrepancies in the reduced solution. In order to fix this oscillatory behavior, we are making use of calibration techniques applying them in this context of RB methods with focus on the steady two dimensional Euler equation around an airfoil.

In the last part of the thesis we focus on MOR methods for parametric nonlinear hyperbolic conservation laws with applications in *Uncertainty Quantification* (UQ). In here, we use the Monte Carlo method to sample the various values of the uncertain parameters, implying a large number of computations. To generate a RB space, we want to find a low dimensional good approximation of the high fidelity functional space. For this, we are using methods as PODEI-Greedy algorithm, by extending the *Empirical Interpolation Method* (EIM) basis functions and the POD-Greedy basis functions in a synchronized way.

Zusammenfassung

Reduzierte-Basis-Methoden (RB-Methoden) können den Rechenaufwand bei wiederholten Simulationen mit begrenzten Rechenressourcen verringern. Sie basieren direkt auf dem hochdimensionalen Modell, das sich aus der Standardformulierung der Finite-Elemente-, Finite-Volumen- oder Finite-Differenzen-Methode ergibt. Die RB-Methoden schränken die Lösung auf einen Unterraum des zugrunde liegenden hochdimensionalen Raumes ein. Der Unterraum wird dabei in einer Trainingsphase durch eine optimal reduzierte Basis bestimmt. Damit wird eine grosse Anzahl (bis zu Millionen) Freiheitsgrade durch nur wenige Koeffizienten repräsentiert, welche in Kombination mit den reduzierten Basisvektoren zu wichtigen rechnerischen Einsparungen führt.

Systeme aus hyperbolischen Erhaltungsgleichungen sind besonders und herausfordernd. In diesem Kontext hängen bewegte Wellen und Diskontinuitäten wie Schocks von verschiedenen Parametern ab und entwickeln sich mit der Zeit. Um die Entwicklung der unstetigen Lösung, welche implizit Nichtlinearitäten beinhaltet, zu erfassen, müssen passende Surrogate entwickelt werden.

Das Ziel dieser Arbeit ist es, innovative Ideen zur Weiterentwicklung der Theorie der Modellordnungsreduktion (MOR) für hyperbolische Erhaltungsgesetze erster Ordnung mit grossen Gradienten und Schocks vorzuschlagen.

Im ersten Teil beweisen wir, dass Modellordnungsreduktion (MOR) unter Verwendung von L^1-Norm-Minimierung auf das Residuum angewendet, zu einer genaueren reduzierten Lösung führt. Die L^1-Norm ist eine natürliche Norm für Evolutionsprobleme, die Unstetigkeiten beinhalten. Um die Kolmogorov N-Breite des Lösungsverteilers zu reduzieren, betrachten wir einen Wörterbuchansatz, der beim Generieren der reduzierten Basisfunktionen auf Kompression verzichtet. Um die Genauigkeit der Methode zu unterstreichen, liefern wir ausserdem robuste Fehlerschätzer für skalare Probleme unter monotonen Verfahren. Wir illustrieren das Verhalten der L^1-Norm-Minimierung basierend auf einem Wörterbuchansatz für lineare und nichtlineare Probleme, ein- sowie auch zweidimensional.

Im zweiten Teil untersuchen wir inwiefern die Parameter in Systemen aus hyperbolischen Erhaltungsgleichungen die Position und Form eines Schocks beeinflussen. Hierbei können Unstimmigkeiten in der reduzierten Lösung auftreten. Um die daraus entstehenden Oszillationen zu beheben, nutzen wir Kalibrierungstechniken, welche wir im Zusammenhang mit RB-Methoden anwenden. Der Schwerpunkt liegt dabei auf der stationären, zweidimensionalen Euler-Gleichung um einen Tragflügel.

Im letzten Teil der Arbeit konzentrieren wir uns auf MOR-Methoden für parametrische, nichtlineare hyperbolische Erhaltungsgleichungen mit Anwendungen in der Unsicherheitsquantifizierung (UQ). Hier verwenden wir die Monte-Carlo-Methode, um die verschiedenen Werte der unsicheren Parameter abzutasten, was eine große Anzahl von Berechnungen impliziert. Um einen RB-Raum zu erzeugen, wollen wir eine niederdimensionale, gute Approximation des High-Fidelity-Funktionsraums finden. Dazu verwenden wir Methoden wie den PODEI-Greedy-Algorithmus, indem wir die Basisfunktionen der empirischen Interpolationsmethode (EIM) und die POD-Greedy-Basisfunktionen synchronisiert erweitern.

Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less. — Marie Curie

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Rémi Abgrall for the continuous support during my Ph.D studies and research, for his patience, enthusiasm, immense knowledge and humor. His guidance helped me throughout all these years of research and during writing this thesis. He is a lifetime role model for me, helping me to become the researcher that I am today. I am also grateful to him for the many opportunities I had during my Ph.D to attend meetings and conferences and to interact with other researchers.

I would like to thank Prof. Yvon Maday and Prof. Charbel Farhat who agreed to be part of my reading committee and also to the members of my doctoral committee.

I am also grateful to Prof. Yvon Maday for inviting me to work with him and his student, Nicolas Cagniart at Pierre and Marie Curie University; to David Amsallem, for the productive discussions in the beginning of my Ph.D; to Svetlana Tokareva, for her insightful ideas and to Davide Torlo, for all his help during my last year of Ph.D.

I am pleased to acknowledge all my friends and colleagues during my years of study as a graduate student at University of Zürich for their friendship and encouragement. In particular, I would firstly like to mention my group: Paola, Tulin, Svetlana, Davide, Maria, Jianfang, Barbara and Élise and my colleagues Céline, Michel, Dario, Stephanie, Wei and Linda. Together with them, we have faced all the ups and downs related to our careers. I would also like to thank my friends in Romania: Cipriana, Roxana and Alexandra.

Finally, I cannot be grateful enough to my parents and to my brother for their constant faith in me. Their love, support and encouragement have given me the strength to achieve my goals. Last but not least, I would like to thank my husband Alex for his unconditional love. I am truly thankful for having you in my life.

Contents

Chapter 1: Introduction

1.1 Motivation

Many engineering applications require the ability to simulate the behavior of a physical system in real-time. This requirement holds in particular when a full parametric exploration of the behavior of the system is sought. In aerodynamics, such an exploration can be done to compute the flow around an aircraft for varying boundary conditions or to design its shape to maximize lift and minimize drag. *Uncertainty Quantification* (UQ) also requires a large number of simulations with varying parameters in order to propagate chaos by means of a Monte-Carlo method or calibrating input parameters by a Markov chain technique. A third important application is flow control.

When such a large number of simulations is required, the cost of one simulation is critical to the application at hand. This cost can be lowered by using sophisticated computer science techniques such as parallelization but such techniques are usually not enough to allow full parametric exploration, especially when computational resources are limited.

In order to tackle this computational time cost issue, over the past four decades *Reduced-Order Modeling* (ROM) have been developed, aiming at replacing the original large-dimension numerical problem by a reduced problem of substantially smaller dimension. Thus, a large number of degrees of freedom (say millions) are represented by only a few number of coefficients in the representation of the full solution in terms of the reduced basis vectors, leading to important computational savings and being capable to operate in near real-time. The most important questions arising in the *Reduced Basis* (RB) methods context are: how can an optimal reduced basis be constructed and how can the evolution of the reduced coefficients be computed in a stable fashion?

A popular method for constructing an "optimal" basis is *Proper Orthogonal Decomposition* (POD), firstly introduced as a tool for the analysis of flows by Lumley [94] and then extended and popularized by Sirovich [127]. The idea behind POD is to collect a few snapshots of the solution and then compute the best approximation of these snapshots in terms of a small number of reduced basis vectors. Mathematically speaking, if $\mathbf{W}_i(t_l) \in \mathbb{R}^m$ denotes the value of the discrete solution $\mathbf{W}$ at grid point $\mathbf{x}_i$, $i = 1, \cdots, N$ and at time t_l, $l = 1, \cdots, N_t$, POD constructs M orthogonal functions $\boldsymbol{\phi}_\ell \in \left[L^2(\mathbb{R}^d)\right]^m$ such that the following functional is minimized:

$$\mathcal{J}(\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_M) = \sum_{l=1}^{N_t} \sum_{i=1}^{Nm} \left\| \mathbf{W}_i(t_l) - \sum_{\ell=1}^{M} \langle \mathbf{W}(t_l), \boldsymbol{\phi}_\ell \rangle \boldsymbol{\phi}_{\ell i} \right\|_2^2, \tag{1.1}$$

where $\boldsymbol{\phi}_{\ell i} \in \mathbb{R}^m$ denotes the value of $\boldsymbol{\phi}_\ell$ at $\mathbf{x}_i$, $\| \cdot \|$ denotes here the Euclidean norm in $\mathbb{R}^m$, and $\langle \cdot , \cdot \rangle$ is the L^2 scalar product. A minimum of the functional $\mathcal{J}$ can be analytically computed by *Singular Value Decomposition* (SVD) and the reduced basis vectors $\boldsymbol{\phi}_\ell$ are the left singular vectors of the snapshots matrix

$$\mathbf{S} = \begin{pmatrix} \mathbf{W}_1(t_1) & \dots & \mathbf{W}_1(t_{N_t}) \\ \vdots & \vdots & \vdots \\ \mathbf{W}_N(t_1) & \dots & \mathbf{W}_N(t_{N_t}) \end{pmatrix}.$$

Defining by $\{\lambda_\ell\}_{l=1}^{N_t}$ the positive eigenvalues of $\mathbf{S}^T\mathbf{S}$ sorted decreasingly, the error associated with the minimum of the functional is

$$\mathcal{J}(\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_M) = \sum_{\ell=M+1}^{N_t} \lambda_\ell. \tag{1.2}$$

In the continuous case, the functions $\boldsymbol{\phi}_\ell(\mathbf{x}) \in \mathbb{R}^m$, are the solution of Fredholm alternative

$$\int_\Omega \mathbf{R}(\mathbf{x}, \mathbf{x}')\boldsymbol{\phi}_\ell(\mathbf{x}')d\mathbf{x}' = \lambda_\ell \boldsymbol{\phi}_\ell(\mathbf{x}), \qquad \text{for all } \mathbf{x} \in \Omega, \tag{1.3}$$

where $\mathbf{R}(\mathbf{x}, \mathbf{x}') = \mathbf{u}(\mathbf{x})\mathbf{u}(\mathbf{x}')^T$.

In both the discrete and continuous cases, the basis dimension M is depending on how fast is the decay of the eigenvalues λ_ℓ. Given a tolerance $\epsilon \ll 1$, M is selected as the smallest dimension such that the following relative truncation error is smaller than ϵ,

$$\frac{\mathcal{J}(\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_M)}{\sum_{l=1}^{N_t}\sum_{i=1}^{Nm} \|\mathbf{W}_i(t_l)\|_2^2} = \frac{\sum_{\ell=M+1}^{N_t} \lambda_\ell}{\sum_{\ell=1}^{N_t} \lambda_\ell}. \tag{1.4}$$

In general, one expects the eigenvalues λ_ℓ to decrease very rapidly to 0. This allows when this assumption is true, to consider only the most energetic modes in the decomposition. Unfortunately, it is not always the case that the eigenvalues λ_ℓ are rapidly converging to zero. This is demonstrated by the following simple counter example for which a simple scalar advection problem defined on $\Omega = [0, 1[$ is considered:

$$\frac{\partial w}{\partial t} + \frac{\partial w}{\partial x} = 0 \tag{1.5a}$$

with the boundary condition

$$w(0, t) = 1 \tag{1.5b}$$

and the initial condition

$$w(x, 0) = 0. \tag{1.5c}$$

The solution is given by a traveling discontinuity

$$w(x, t) = \begin{cases} 1 & \text{if } x \le \min(t, 1) \\ 0 & \text{otherwise.} \end{cases}$$

Considering grids $x_i = i/N$, $i = 0, \dots, N$ for varying number of grid points N and snapshots collected at times as $t_k = k\Delta t$, with $\Delta t = 1/N$, a series of POD bases are constructed numerically. For each grid size N, the eigenvalues $\lambda_\ell(N)$ are reported in Figure 1.1. One can observe that the ratio $\lambda_\ell(N)/\lambda_1(N)$ behaves like $1/k$.

This problem highlights one important difficulty in the *Model Order Reduction* (MOR) of hyperbolic problems namely, the assessment of whether such a problem is reducible or not before attempting to reduce it. It shows also that it is difficult to select only a few dominant modes, due to the slow decay of the POD eigenvalues. However, after the first 10% of the reported singular values, the energy is shown to have been reduced by almost three orders of magnitude but this does not imply as whether this suffices to perform MOR within a reasonable accuracy ϵ (after all, MOR is the science of trading some of the accuracy for a lot of speed).

Nevertheless, even if the RB technique itself has relatively rarely been applied to first-order hyperbolic problems, there is a vast literature on developing and applying POD- based and other techniques for the MOR of CFD problems, as [12, 14, 20, 37, 98, 118, 134, 139], to name just a few. In this thesis, we will present an alternative to the POD method for constructing *Reduced Order Basis* (ROB) namely, a dictionary approach, which doesn't include any compression.

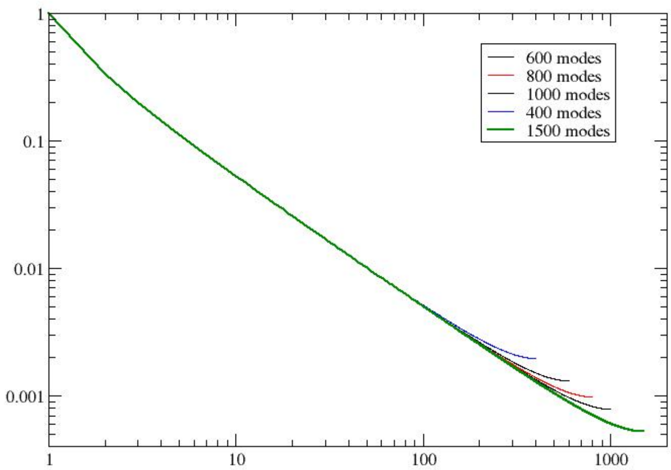


Figure 1.1: *In log-log coordinate, plot of the ratio of POD eigenvalues* $\log(\lambda_k(N)/\lambda_1(N))$ *for* $N = 400, 600, 800, 1000, 1500$ *grid points.*

Concerning compressible fluids, there is another difficulty. In problem (1.3), one needs a norm. In the case of incompressible flows, a natural norm is related to the kinetic energy. For compressible materials, however, one needs to take into account the density, velocity and the energy, i.e. the thermodynamics. A simple L^2-norm cannot be used because one cannot combine in a quadratic manner these variables, for dimensional reasons. Only a non-dimensionalization of the variables can alleviate the dimensionality issue [14, 37]. The natural equivalent of the L^2-norm is however related to the entropy, which is not quadratic: if a minimization problem can be set up, its solution is non trivial. These arguments were raised in [118], and an energy-based norm was developed in [20] for linearized compressible flows.

In addition to the reduced basis choice, a key ingredient in projection-based model reduction is the definition of the reduced system of equations. For symmetric systems such as those arising in elliptic and parabolic PDEs, Galerkin projection is the method of choice. For non-symmetric systems, it has been shown that minimizing the L^2-norm of the residual is preferable for stability, unicity and optimality considerations [36, 37]. Nevertheless, in this thesis, we will present a minimum residual approach for determining the generalized coordinates by using the L^1-norm (adding a regularization and a perturbation term to it) as an alternative to the L^2-norm and we will show it's robustness in the context of hyperbolic problems.

In the context of MOR, numerous methods such as reduced basis and its generalization, proper orthogonal decomposition, generalized empirical interpolation, rely on the implicit assumption that the solution manifold that gathers the solution of a PDE as parameters vary, can be well approximated by low dimensional spaces. But how well the solution manifold $\mathcal{M}_{\mathcal{D}}$ can be approximated by a finite-dimensional subspace of a prescribed dimension? This is related to the notion of *Kolmogorov N-width* of solution manifold, which is defined as:

$$d_N(\mathcal{M}_{\mathcal{D}}, X) = \inf_{E_N} \sup_{f \in \mathcal{M}_{\mathcal{D}}} \inf_{g \in E_N} ||f - g||_X,$$

where X some normed linear space in which $\mathcal{M}_{\mathcal{D}}$ is embedded and E_N represents all linear subspaces embedded in X. So in practice, one needs to build an algorithm to find subspaces close to the optimal ones given by the Kolmogorov n-width i.e $d_N(\mathcal{M}_{\mathcal{D}}, X)$ is small. The main issue for hyperbolic problems is that the Kolmogorov N-width of the solution manifold will not have the good decay properties required for standard ROM, so is not amenable to linear approximation, even though the structure is simple. This is also the case of the steady 2D Euler equation around an airfoil, which for different parameter inputs (Mach numbers and angle of attacks), the positions and the shapes of the shock are varying (see Figure 1.2). Some other classical and simple example illustrating the limitations of the reduced models due to large Kolmogorov N-width is the pure linear convection with constant velocity. One option to fix the N-width issue related to scalar conservation laws depending on a set of parameters is to adapt shock fitting ideas in the context of ROM [128]. Unfortunately, this method, just as any other shock fitting method, is somehow limited to one dimensional problems. Another way to fix this N-width issue is to use a preconditioning step such as calibration [34]. In [34] it is shown that in the 1D case, the corresponding calibrated solution manifold will have a smaller Kolmogorov N-width than the initial solution manifold. We would try to adapt the same strategy for our 2D Euler equation in order to fix the Kolmogorov N-width decay issue and for enhancing the robustness of MOR for CFD problems with shocks.

Of a great importance is also the study of the uncertainties in the input parameters of the PDEs, especially for hyperbolic conservation laws, as the case of elliptic problems is well studied [39–41]. In practice, the input parameters are obtained by measurements (observations) and these measurements are not always very precise, involving some degree of uncertainty [26, 54]. A good example of hyperbolic conservation laws is when computing the flow past an airfoil or a wing, the inputs for this calculation, such as the inflow Mach number, the angle of attack, as well as the parameters that specify the airfoil geometry, are all measured with some uncertainty. This uncertainty in the inputs results in the propagation of uncertainty in the solution [9]. Moreover, the need of MOR for UQ is obvious by just taking into

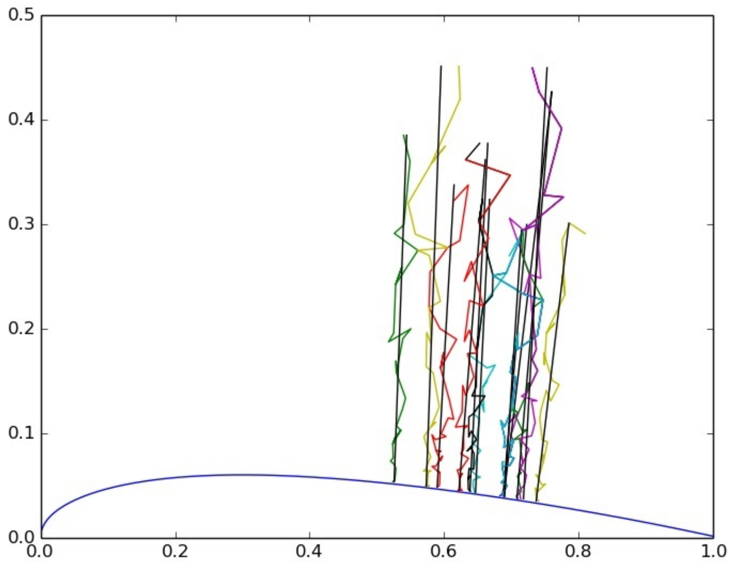


Figure 1.2: Position of the shock for various AoA and Mach numbers

account that these problems feature high-dimensionality, low regularity and arbitrary probability measures. However, the classical methods (Monte Carlo, stochastic Galerkin projection method, stochastic collocation method, etc) can not be directly applied to solve the underlying deterministic PDEs, since they might need millions of full solutions (or even more), in order to achieve a certain accuracy. Other challenge when developing ROM algorithms for quantifying uncertainty in solutions of conservation laws with random inputs is dealing with unsteady nonlinear problems, which involve discontinuities. Hence, robust RB methods have to be developed, which together with an a posteriori error estimate to be able to deal with the non-linear terms and to capture the evolution of the discontinuous solutions.

1.2 Strategy and Objectives

The main idea of this dissertation is to develop robust MOR methods which are adapted to hyperbolic conservation laws, in order to circumvent all those issues presented in Section 1.1. This idea was firstly motivated by example (1.5) presented in Section 1.1, where another strategy which does not imply compression will be use. In this thesis, an approach based on a dictionary of solutions [99] is presented, as an alternative to using a truncated reduced basis based on POD. The elements of this dictionary are solutions $\mathbf{W}(t_l; \boldsymbol{\mu}_j)$ computed for varying values of time t_l and parameters $\boldsymbol{\mu}_j \in \mathbb{R}^p$. In this case, each solution is considered to be a reduced basis vector. In turn, localization in time and space can be easily enforced by only considering basis vectors corresponding to restricted sub-domains of the time and parameters spaces. In addition to the reduction in number of basis vectors, in this thesis we demonstrate that a key advantage of a dictionary approach is a better approximation of the states having sharp gradients and discontinuities. In particular, it shows that avoiding basis truncation such as the one occurring in *Proper Orthogonal Decomposition* (POD) [127] or Non-Negative Matrix Factorization [18] avoid Gibbs phenomenon.

In addition to choosing an appropriate dictionary, selecting an approach for computing a reduced solution based on that dictionary is also crucial. For symmetric systems such those arising in elliptic and parabolic PDEs, Galerkin projection is a natural approach but there is no motivation for using Galerkin projection for nonlinear compressible flows. However, for non-symmetric systems, strategies based on the minimization of the residual arising from the reduced approximation have been successfully developed for compressible flows in [32, 37, 87] but all these approaches rely on the minimization of the residual in L^2-norm. Based on these ideas of minimizing the residual and on the work of Lavery [83] and Guermond et al [63, 64], who showed that in the case of hyperbolic problems, the numerical solution can retain an excellent non-oscillatory behavior by only minimizing the L^1-norm of the PDE residual, we propose model order reduction methods based on the minimization of the L^1-norm of the residual and we demonstrate its advantage in conjunction with a dictionary approach for reducing problems with sharp gradient and shocks. Moreover, we show that this norm is closely linked to the concept of weak solutions of hyperbolic conservation laws.

Another objective of this thesis is to take into account the behavior of the moving shocks and discontinuities presented in Section 1.1, which imply changes in their position and shapes. This issue and the ones presented in Section 1.1 related to this problem, were a strong motivation to start a joint work with Prof. Yvon Maday and Nicolas Cagniart from University Pierre and Marie Curie. Thus, we are making use of calibration ideas [34], which in relation with MOR and L^1-norm minimization techniques will constitute a complete framework when dealing with these problems. The idea of calibration is simple. Firstly, consider a family of mappings and in an offline phase, use these mappings to calibrate the offline computed solutions in order to get a reduced basis as small as possible, i.e to reduce the Kolmogorov N-width of the solution manifold. In this step, the calibration is achieved by locating the position of the shock and the family of mappings will be constructed using a Gordon-Hall (G-H) type of mapping [59,91,92]. Secondly, construct in an online phase, a cheap reduced scheme approximating the truth solver, which uses the calibrated manifold from step one and L^1-minimization ideas. However, the development of such an online phase is not a straightforward task. Considering the existence of a fully functioning Computational Fluid Dynamics (CFD) code, we need to recast the original problem defined on a physical domain, onto an equivalent problem defined on a reference domain, which relies on the variational form of the PDE at hand. This is a well studied problem in the elliptic and parabolic communities [93, 114] but a similar procedure for our hyperbolic conservation laws will only lead to a non conservative formulation and then the use of the CFD code it will not be possible. Making use of Piola transformation, we will show that only modifying the flux and the boundary conditions will lead to a conservative formulation which fits in the CFD code and which will allow the computation of the reduced solution.

The last objective of this dissertation is to reduce the computational cost of the hyperbolic conservation laws with random inputs and is a joint work with Davide Torlo and Dr. Svetlana Tokareva from University of Zürich. Because we are interested in MOR for unsteady non-linear hyperbolic systems of conservation laws, which involves sharp gradients and shocks, we need to explore the parameter-time framework but in the same time, also to deal with nonlinearities. For this, we are using a POD–EIM–Greedy algorithm [50], which is a combination of different algorithms, such as POD [75, 82], POD–greedy [69], EIM–greedy [21]. The idea is to use a POD–Greedy algorithm (POD algorithm in time and a Greedy algorithm in the parameter space), which deals with unsteady problems in the reduced basis context and with the help of which we will construct the reduced basis. And also to use the *Empirical Interpolation Method* (EIM) algorithm which deals with non-linearities i.e approximates a general non-linear function by a sum of affine terms by means of interpolation. And in the end, synchronize at each step of the main greedy algorithm the extension of the EIM basis function and the one of POD–Greedy basis functions.

1.3 Thesis Accomplishments and Outline

The major contributions of this dissertation are as follows:

- A robust MOR method based on L^1-norm minimization of the residual and approximation via dictionaries in one dimensional space (1D) and two dimensional space (2D) for steady and unsteady nonlinear hyperbolic systems of conservation laws including a proof of the sparsity of the reduced solution, the development of robust error estimators for the scalar case and a discussion on the cost of the method using hyper-reduction;
- A robust MOR method for nonlinear hyperbolic systems of conservation laws using calibration with focus on the steady two dimensional Euler equation around an airfoil. This approach includes an offline calibration procedure, the construction of a family of mappings, a fully functioning reduced scheme, an offline-online decomposition based on L^1-norm minimization, description of hyper-reduction ideas and illustration of numerical experiments that serve as a proof of concept for the global method;
- A robust MOR method for unsteady nonlinear hyperbolic systems of conservation laws with applications in UQ including the derivation of an error indicator and applications in 1D and 2D for problems with random inputs.

This dissertation is composed of three different papers and is organized as follows. Chapter 2 introduces a formal mathematical background on hyperbolic systems of conservation laws and reduced basis methods for parametrized PDEs. A robust MOR using L^1-norm minimization and approximation via dictionaries is proposed in Chapter 3. We develop the method and analyze it in the simplified one dimensional case. We show in this case that error bounds with the full model can be obtained provided that a suitable minimization approach is chosen. The capability of the algorithm is then shown on nonlinear scalar problems, one dimensional unsteady fluid problems and two dimensional steady compressible problems. Another contribution of this chapter is the discussion of hyper-reduction ideas in this context and on the cost of the method. In Chapter 4, adapted standard MOR techniques for hyperbolic problems are presented. More precisely, in this chapter we propose a complete framework of the calibration procedure that allows to use standard ROM techniques to solve the two dimensional Euler equation around an airfoil. First, an offline calibration procedure that reduces the Kolmogorov N-width of the solution manifold is studied and then, a cheap reduced scheme approximating the truth solver is presented. It uses the L^1-norm minimization technique and the calibrated manifold constructed in the offline phase. We discuss its computational complexity and finally, we present numerical simulations that illustrate the overall feasibility of the method. Chapter 5 presents reduced basis techniques applied on conservation laws with random input parameters. We provide an error indicator which under some hypothesis is also an error upper bound for the difference between the high fidelity solution and the reduced one. Numerical results for the stochastic unsteady non-linear hyperbolic conservation laws with random data in both 1D and 2D are also presented. Finally, conclusions and indications for future research are provided in Chapter 6.

Chapter 2: Preliminaries

2.1 Hyperbolic systems of conservation laws

2.1.1 Introduction to hyperbolic systems of conservation laws

A *system of m balance laws* with $m \geq 1$ on a d-dimensional (d=1,2,3) Lipschitz domain $\Omega \subseteq \mathbb{R}^d$ and on a time domain $\mathbb{R}_+ = \{t \in \mathbb{R} : t \geq 0\}$ is described as:

$$\begin{cases} \mathbf{W}_t(\mathbf{x}, t) + \operatorname{div}(\mathbf{f}(\mathbf{W})) & = \mathbf{S}(\mathbf{x}, t, \mathbf{W}, \nabla\mathbf{W}),\ (\mathbf{x}, t) \in \Omega \times \mathbb{R}_+, \\ \mathbf{B}(\mathbf{W}) & = \mathbf{g}(\mathbf{x}, t),\ (\mathbf{x}, t) \in \partial\Omega \times \mathbb{R}_+, \\ \mathbf{W}(\mathbf{x}, t = 0) & = \mathbf{W}_0(\mathbf{x}),\ \mathbf{x} \in \Omega, \end{cases} \tag{2.1}$$

where $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$ is the space variable ($1 \leq d \leq 3$), $t \in \mathbb{R}_+$ is the time variable, $\partial\Omega$ is the boundary of the domain and $\mathbf{W}_t$ denotes the time derivative $\partial/\partial t$ of the physical variables

$$\mathbf{W} = \mathbf{W}(\mathbf{x}, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^m,\ m \geq 1.$$

The system is called a *conservation law* if the source term $\mathbf{S} = 0$.

The flux

$$\mathbf{f} : \mathbb{R}^m \to (\mathbb{R}^m)^d,\ \mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_d)$$

is a collection of directional vector-valued flux functions, where

$$\mathbf{f}_i(\mathbf{W}) : \mathbb{R}^m \to \mathbb{R}^m,\ i = 1, \cdots, d$$

are sufficiently smooth, i.e Lipschitz continuous[1] with respect to the state variable.

The divergence is a differential operator corresponding to the spatial domain Ω,

$$\operatorname{div} := \sum_{i=1}^{d} \frac{\partial}{\partial x_i}$$

and the source term is given by:

$$\mathbf{S} : \Omega \times \mathbb{R}_+ \times \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m.$$

To the partial differential equation (PDE) are added the boundary conditions $\mathbf{g}$, which are imposed through a suitable boundary operator $\mathbf{B}$,

$$\mathbf{g} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^m$$

and the initial conditions are given by:

$$\mathbf{W}_0 : \Omega \to \mathbb{R}^m.$$

[1] A function $f : X \to Y$ is called Lipschitz continuous if there exists a constant C such that for each $x, y \in X$ one has: $|f(x) - f(y)| \leq C|x - y|$

2.1.2 Examples of conservation laws equations

2.1.2.1 Burgers' equation

Consider the *viscid Burgers' equation* [33] to be the nonlinear parabolic PDE:

$$w_t^\epsilon + w^\epsilon w_x^\epsilon = \epsilon w_{xx}^\epsilon, \tag{2.2}$$

where ϵw_{xx}^ϵ is a viscous term. This is the simplest scalar problem ($m = 1$) which can incorporate both the nonlinear propagation effects and the diffusive effects. When $\epsilon \to 0$ in equation (2.2) (i.e, $w^\epsilon \to w$ as $\epsilon \to 0$), we obtain the following *inviscid Burgers' equation*:

$$w_t + f(w)_x = 0, \quad f(w) = \frac{w^2}{2}, \; x \in \mathbb{R}, \; t \in \mathbb{R}_+, \tag{2.3}$$

which is a nonlinear scalar equation of type (2.1) with $d = 1$.

Hence, there is an important connection between the viscid Burgers' equation (2.2) and its inviscid counterpart (2.3) namely, equation (2.3) is the limit of (2.2) as $\epsilon \to 0$. This will be discussed in Section 2.1.4, Theorem 2.1.6. Nevertheless, in order to stay in the structure of hyperbolic conservation laws, we will consider in our applications a viscosity-free equation ($\epsilon = 0$) but which models many physics of fluid dynamics.

2.1.2.2 Euler equations

A good example of a nonlinear system of conservation laws is the *Euler equations* of gas dynamics, which describe the time evolution of mass density, velocities and pressure in compressible fluids. In the general case, when $d \geq 1$ and $m = 3$, Euler equations express respectively the laws of conservation of mass, momentum and total energy for the fluid and writes:

$$\begin{cases} \rho_t + div(\rho\mathbf{w}) & = 0, \\ (\rho\mathbf{w})_t + div(\rho\mathbf{w} \otimes \mathbf{w} + p\mathbf{I}) & = 0, \\ E_t + div((E + p)\mathbf{w}) & = 0, \end{cases} \tag{2.4}$$

or rewritten in the following form

$$\mathbf{W}_t + \text{div } \mathbf{f}(\mathbf{W}) = 0,$$

where $\mathbf{W} = \begin{pmatrix} \rho \\ \rho\mathbf{w} \\ E \end{pmatrix}$ and the flux $\mathbf{f}(\mathbf{W}) = \begin{pmatrix} \rho\mathbf{w} \\ \rho\mathbf{w} \otimes \mathbf{w} + p\mathbf{I} \\ (E + p)\mathbf{w} \end{pmatrix}$. In this case, ρ represents the density of the fluid and $\mathbf{w}$ the velocity field (when $d = 3$, $\mathbf{w} = (w_1, w_2, w_3)$). The total energy E and the pressure p are related by the ideal gas equation of state:

$$p = (\gamma - 1)(E - \frac{1}{2}\rho|\mathbf{w}|^2), \tag{2.5}$$

where γ represents the ratio of specific heats, which for air, in standard day conditions, equals 1.4. In 1D ($d = 1,\ m = 3$), Euler equations can be written in form of (2.1) as:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho w_1\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho w_1\\ \rho w_1^2 + p\\ w_1(E+p)\end{pmatrix} = 0 \tag{2.6}$$

and in 2D ($d = 2,\ m = 4$), we obtain:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho w_1\\ \rho w_2\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho w_1\\ \rho w_1^2 + p\\ \rho w_1 w_2\\ w_1(E+p)\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho w_2\\ \rho w_1 w_2\\ \rho w_2^2 + p\\ w_2(E+p)\end{pmatrix} = 0, \tag{2.7}$$

where (w_1, w_2) is the 2D fluid velocity. Euler equations are used in aircraft designing, gas turbines, flow modeling and many other applications.

2.1.2.3 Flow through a nozzle in 1D

A nozzle is an extremely efficient device for converting thermal energy into kinetic energy. Nozzles arise in a vast range of applications. Obvious ones are the thrust nozzles of rocket and jet engines. Converging-diverging ducts also come up in aircraft engine inlets, wind tunnels and in all sorts of piping systems designed to control gas flow.

An ideal gas flowing through a straight nozzle with a slowly-varying cross-sectional area $A = A(x)$, where x measures the distance along the nozzle, can be seen as a particular case of Euler equations, namely, *quasi-1D Euler equations* ($d = 1,\ m = 3$), which can be written in form of (2.1) as:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = \mathbf{S}, \tag{2.8}$$

where $\mathbf{W} = \begin{pmatrix}\rho A\\ \rho w_1 A\\ EA\end{pmatrix}$, $\mathbf{f} = \begin{pmatrix}\rho w_1 A\\ (\rho w_1^2 + p)A\\ w_1(E+p)A\end{pmatrix}$ and $\mathbf{S} = \begin{pmatrix}0\\ p\frac{\partial A}{\partial x}\\ 0\end{pmatrix}$.

A nice application in the steady case is the converging-diverging nozzle, which has direct applications for the jet engines. The usual configuration is the following: gas flows through the nozzle from a region of high pressure (usually referred to as the chamber) to one of low pressure (referred to as the ambient or tank). The chamber is usually big enough so that any flow velocities here are negligible. Gas flows from the chamber into the converging portion of the nozzle, past the throat, through the diverging portion and then exhausts into the ambient as a jet. The pressure of the ambient is also called the "back pressure". In studying this problem, a very important role has the *Mach number*, which represent the ratio of flow velocity past a boundary to the local speed of sound. In a steady internal flow (like a nozzle), the Mach number can only reach 1 at a minimum in the cross-sectional area. The flow pattern will differ depending on the back pressure and this is when the flow is "choked". As the back pressure is lowered below than needed to just choke the flow, a region of supersonic flow forms just downstream of the throat. Unlike a subsonic flow, the supersonic flow accelerates as the area gets bigger. This region of supersonic acceleration is terminated by a normal shock wave. The shock wave produces a near-instantaneous deceleration of the flow to subsonic speed. This subsonic flow then decelerates through the remainder of the diverging section and exhausts as a subsonic jet. In this regime, if the back pressure is lowered or raised, then the length of the supersonic flow in the diverging section, before the shock wave, is increased or decreased (see Figure 2.1).

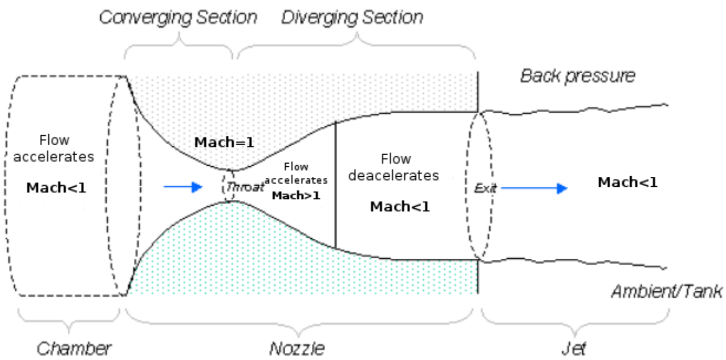


Figure 2.1: Shock in converging-diverging nozzle

2.1.3 Weak solution and Rankine-Hugoniot condition

For simplicity, we consider for the rest of this section the case when $\Omega = \mathbb{R}^d$, which leads to the simplest example of a system of hyperbolic conservation laws, namely the *Cauchy problem* or, *initial value problem* (IVP):

$$\frac{\partial \mathbf{W}(\mathbf{x},t)}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x_i}\mathbf{f}_i(\mathbf{W}) = 0, \;\; (\mathbf{x},t) \in \mathbb{R}^d \times \mathbb{R}_+ \tag{2.9}$$

$$\mathbf{W}(\mathbf{x}, t=0) = \mathbf{W}_0(\mathbf{x}), \;\; \mathbf{x} \in \mathbb{R}^d. \tag{2.10}$$

In this case, there is no need to specify the boundary condition $\mathbf{g}$, as $\partial\Omega = \emptyset$.

It is well known that the Cauchy problem for nonlinear hyperbolic systems is facing two major challenges. First, even when starting from a smooth initial condition $\mathbf{W}_0$, the classical solutions may be visualized as propagating waves which in finite time T become steeper, developing jump discontinuities which propagate on as shocks (see Figure 2.2). Hence, it becomes imperative to introduce the notion of *weak solutions* to systems of conservation laws i.e the solutions of (2.9)-(2.10) in the sense of distributions [56, 84, 88].

Definition 2.1.1 (Weak solution). *Assume that $\mathbf{W}_0 \in \mathbf{L}^\infty_{loc}(\mathbb{R}^d)$, where $\mathbf{L}^\infty_{loc}$ is the space of locally bounded measurable functions and let $\mathbf{C}^1_0(\mathbb{R}^d \times \mathbb{R}_+)$ being the space of $\mathbf{C}^1$ functions with compact support in $\mathbb{R}^d \times \mathbb{R}_+$. A locally integrable function $\mathbf{W} \in \mathbf{L}^\infty_{loc}(\mathbb{R}^d \times \mathbb{R}_+)$ is called a weak solution of the Cauchy problem (2.9)-(2.10) if all smooth test functions $\boldsymbol{\varphi} \in \mathbf{C}^1_0(\mathbb{R}^d \times \mathbb{R}_+)$ are satisfying the following identity:*

$$\int_0^\infty \int_{\mathbb{R}^d} \mathbf{W} \cdot \boldsymbol{\varphi}_t \; d\mathbf{x} \; dt + \int_0^\infty \int_{\mathbb{R}^d} \sum_{i=1}^{d} \mathbf{f}_i(\mathbf{W}) \cdot \boldsymbol{\varphi}_{x_i} \; d\mathbf{x} \; dt + \int_{\mathbb{R}^d} \mathbf{W}(\mathbf{x},0) \cdot \boldsymbol{\varphi}(\mathbf{x},0) \; d\mathbf{x} = 0, \tag{2.11}$$

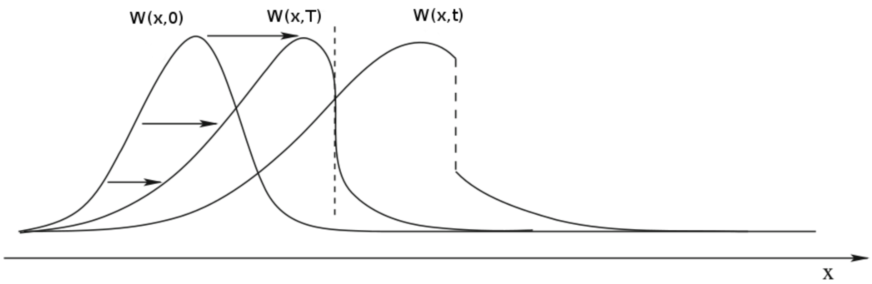


Figure 2.2: The wave propagation speed depends on $\mathbf{W}$, so the profile of the solution changes in time, eventually leading to shock formation at a finite time T

where the dot $\cdot$ represents the Euclidean inner product and the differential operators $\frac{\partial}{\partial t}$ and $\frac{\partial}{\partial x_i}$ are applied component-wise:

$$\boldsymbol{\varphi}_t = (\frac{\partial}{\partial t}\varphi_1, \cdots, \frac{\partial}{\partial t}\varphi_p),\ \boldsymbol{\varphi}_{x_i} = (\frac{\partial}{\partial x_i}\varphi_1, \cdots, \frac{\partial}{\partial x_i}\varphi_p).$$

So, the idea behind the weak solution of a system of conservation laws is very simple: multiply the PDE with a smooth test function, integrate over the space-time domain $\mathbb{R}^d \times \mathbb{R}_+$, and finally, integrate by parts. In this case, this results in having no derivatives on $\mathbf{W}$, hence requiring less smoothness.

It is clear until this point that any classical solution of (2.9)-(2.10) is also a weak solution. Moreover, the weak solutions don't need to be differentiable or continuous in order to satisfy (2.11) and may not be classical solutions of (2.9)-(2.10) and therefore admit discontinuities. Let's consider that $\mathbf{W}$ is a weak solution of (2.9)-(2.10) which is discontinuous across the curve $\mathbf{x} = \boldsymbol{\xi}(t)$ but $\mathbf{W}$ is smooth everywhere else. Let $\mathbf{W}_-(\mathbf{x}, t)$ be the limit of $\mathbf{W}$ approaching $(\mathbf{x}, t)$ from the left and let $\mathbf{W}_+(\mathbf{x}, t)$ be the limit of $\mathbf{W}$ approaching $(\mathbf{x}, t)$ from the right (see Figure 2.3).

Therefore, the following result provides the necessary and sufficient conditions such that a *piecewise continuous function* to be a weak solution of (2.9)–(2.10) [57].

Theorem 2.1.2. *Consider a system of conservation laws (2.9)-(2.10) on $\Omega = \mathbb{R}^d$. A piecewise C^1 function $\mathbf{W} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^m$ is a weak solution of (2.9)-(2.10) if and only if the following two conditions are fulfilled:*

1. $\mathbf{W}$ *is a classical solution of (2.9) in the subdomains where* $\mathbf{W}$ *is* C^1*;*
2. $\mathbf{W}$ *satisfies the Rankine-Hugoniot jump condition across the discontinuity* $\mathbf{x} = \xi(t)$*,*

$$(\mathbf{W}_+ - \mathbf{W}_-)n_t + \sum_{i=1}^{d} (\mathbf{f}_i(\mathbf{W}_+) - \mathbf{f}_i(\mathbf{W}_-))n_{x_i} = 0, \tag{2.12}$$

where $\mathbf{n} = (n_t, n_{x_1}, \cdots, n_{x_d})$ *is the unit normal to the surface of the discontinuity* $\mathbf{x} = \xi(t)$*.*

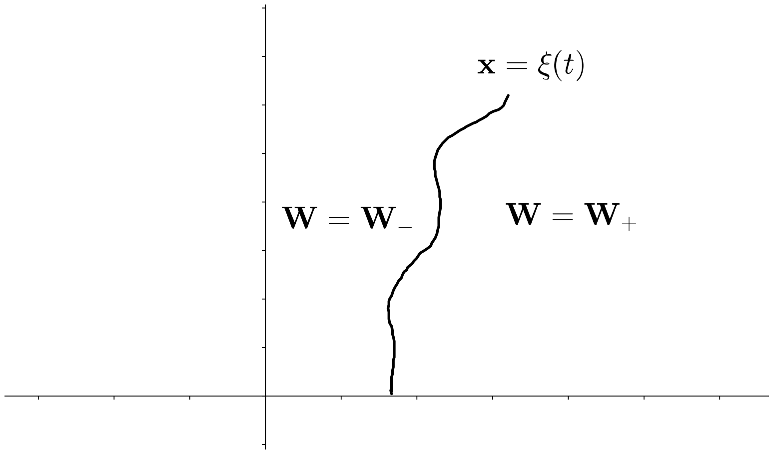


Figure 2.3: Discontinuity of the solution $\mathbf{W}$

Denote by

$$[\mathbf{W}] = \mathbf{W}_+ - \mathbf{W}_- \tag{2.13}$$

the jump of $\mathbf{W}$ across the discontinuity ξ and by

$$[\mathbf{f}_i(\mathbf{W})] = \mathbf{f}_i(\mathbf{W}_+) - \mathbf{f}_i(\mathbf{W}_-),\ i = 1, \cdots, d \tag{2.14}$$

the jump of the flux $\mathbf{f}_i(\mathbf{W})$ across the discontinuity ξ. Hence, the equation (2.12) can be rewritten as:

$$n_t[\mathbf{W}] + \sum_{i=1}^{d} n_{x_i}[\mathbf{f}_i(\mathbf{W})] = 0. \tag{2.15}$$

If $(n_{x_1}, \cdots, n_{x_d}) \neq (0, \cdots, 0)$, then we can consider the normal vector in the following form:

$$\mathbf{n} = \begin{pmatrix} -s \\ \boldsymbol{\eta} \end{pmatrix},$$

where $s \in \mathbb{R}$ and $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)^T$ is a unit vector in $\mathbb{R}^d$. Then (2.15) can be written in the following form:

$$s[\mathbf{W}] = \sum_{i=1}^{d} \eta_i[\mathbf{f}_i(\mathbf{W})],$$

where s represents the *shock speed.*

2.1.4 Entropy solutions

The second challenge is related to the fact that the weak solutions of a system of conservation laws are not unique. For example, consider the 1D Burgers' equation (2.3) with the initial data

$$w_0(x) = \begin{cases} 1, & \text{if} \quad x \geq 0, \\ 0, & \text{if} \quad x < 0. \end{cases} \tag{2.16}$$

As shown in Figure 2.4, for every $0 < \alpha < 1$, a weak solution is

$$w(x,t) = \begin{cases} 0, & \text{if} \quad x < \alpha t/2, \\ \alpha, & \text{if} \quad \alpha t/2 \leq x < (1+\alpha)t/2, \\ 1,, & \text{if} \quad x \geq (1+\alpha)t/2. \end{cases} \tag{2.17}$$

Indeed, the piecewise constant function w trivially satisfies the equation outside the jumps. Moreover, the Rankine-Hugoniot conditions hold along two lines of discontinuity $\{x = \alpha t/2\}$ and $\{x = (1+\alpha)t/2\}$, for all $t > 0$.

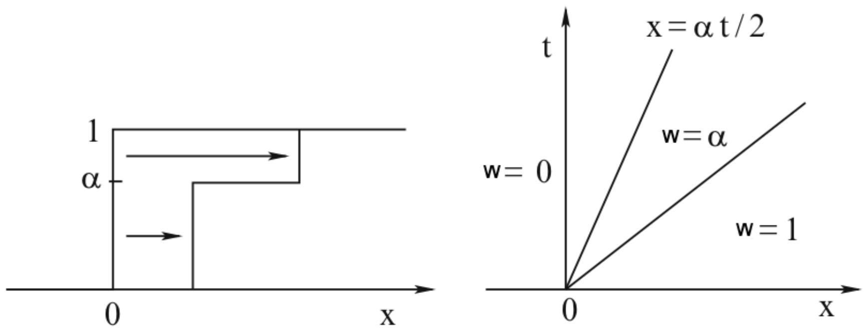


Figure 2.4: For every $\alpha \in [0,1]$ one obtains a different weak solution of Burgers' equation, always with the same initial data

Thus, we have to impose some conditions directly on the weak solutions in order to pick the correct one from a physical point of view. In theory, in order to solve this issue of non-uniqueness, one can introduce a diffusive term into the equations to obtain a system of equations with a unique smooth solution, and then let the coefficient of this term go to zero. This method of *vanishing viscosity* require the evaluation of a complicated system of equations. For this reason, one can derive other conditions that can be imposed on weak solutions and which are easier to be checked. These conditions are called *entropy conditions* and are sufficient to recognize precisely those discontinuities that are physically correct and specify a unique solution [88].

Consider the Cauchy problem (2.9)-(2.10) for scalar conservation laws, i.e $m = 1$ and we denote the single conserved variable by $w(\mathbf{x}, t)$, the initial condition as $w_0(\mathbf{x})$ and the flux function with $f : \mathbb{R} \to \mathbb{R}^d$. The resulting Cauchy problem ($\Omega = \mathbb{R}^d$) writes:

$$w_t + \operatorname{div}(f(w)) = 0, \; \forall (\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+ \tag{2.18}$$

$$w_0(\mathbf{x}, 0) = w_0(\mathbf{x}), \; \forall \mathbf{x} \in \mathbb{R}^d. \tag{2.19}$$

Definition 2.1.3. *A convex function $E : \mathbb{R}^m \to \mathbb{R}$ is called an entropy if there exist d functions $F_i : \mathbb{R}^m \to \mathbb{R}, \; i = 1, \ldots, d$ called entropy fluxes which satisfy the following relation:*

$$F_i'(\mathbf{W}) = E'(\mathbf{W}) \mathbf{f}_i'(\mathbf{W}), \; 1 \leq i \leq d, \tag{2.20}$$

where we denote for simplicity

$$E' = \nabla E^T = \Big(\frac{\partial E}{\partial W_1}, \cdots, \frac{\partial E}{\partial W_m}\Big), \quad F' = \nabla F^T = \Big(\frac{\partial F}{\partial W_1}, \cdots, \frac{\partial F}{\partial W_m}\Big)$$

and the linear mappings $\mathbf{f}_i^{'} : \mathbb{R}^m \to \mathbb{R}^m$ *with the matrix*

$$\mathbf{f}_i^{'} = \Big(\frac{\partial f_{ji}}{\partial W_k}\Big), \quad 1 \leq j, k \leq m.$$

In the case of a scalar conservation law of type (2.18), equation (2.20) writes

$$F_i^{'}(w) = E'(w) f_i^{'}(w), \ 1 \leq i \leq d. \tag{2.21}$$

The pair (E,F) is called entropy pair. From (2.21), we can deduce that the entropy functions are considered to continuously differentiable.

Remark 2.1.4. *In this context, of a great importance for the scalar problems are the so-called Kružkov entropy pairs [80] with the entropy functions*

$$E(w) = |w - k|, \ k \in \mathbb{R} \tag{2.22}$$

and their corresponding entropy fluxes defined as

$$F_i(w) = sgn(w - k)(f_i(w) - f_i(k)), \ i = 1, \cdots, d, \ k \in \mathbb{R}. \tag{2.23}$$

The Kružkov entropy functions (2.22) are a generalization of the entropy functions defined in Definition 2.1.3, as they are not of class C^1.

Any classical solution $\mathbf{W}$ of (2.9)-(2.10) satisfies the following conservation law:

$$E(\mathbf{W})_t + \sum_{i=1}^{d} \frac{\partial}{\partial x_i} F_i(\mathbf{W}) = 0. \tag{2.24}$$

Remark 2.1.5. *In this subsection, the Cauchy problem for scalar conservation laws was considered because when* $m = 1$*, any convex function* E *is an entropy and the entropy fluxes* F_i *are determined as the primitive of the* $E' f_i^{'}$. *Anyway, for a more general case (*$m > 1$*), finding the entropy functions is a much more difficult task because there often exists only a limited number or even only one single entropy (the physical one), because the corresponding compatibility relations are much more restrictive (see [43]).*

The natural question based on the previous results would be if there exists also a relation between the weak solutions $\mathbf{W}$ of (2.9)-(2.10) and the entropy pair (E, F).

Given a small parameter $\epsilon > 0$, consider the following viscous parabolic system, associated with the nonlinear conservation law (2.9):

$$\mathbf{W}_t^{\epsilon} + \operatorname{div} \mathbf{f}(\mathbf{W}^{\epsilon}) = \epsilon \Delta \mathbf{W}^{\epsilon}, \tag{2.25}$$

where $\epsilon \Delta \mathbf{W}^{\epsilon}$ can be viewed as a *viscosity term.* This perturbation transforms the system of conservation laws (2.9)-(2.10) into an advection-diffusion equation which is known to possess smooth solutions $\mathbf{W}^{\epsilon} \in C^{\infty}(\mathbb{R}^d \times \mathbb{R}_+)$.

Multiplying the equation (2.25) with $E'(\mathbf{W}^{\epsilon})$, where E is a C^2 entropy function, applying (2.20) and using the chain rule successively, based on [25, 43, 57], we obtain:

$$E'(\mathbf{W}^{\epsilon}) \cdot \frac{\partial}{\partial t} \mathbf{W}^{\epsilon} + \sum_{i=1}^{d} E'(\mathbf{W}^{\epsilon}) \cdot \frac{\partial}{\partial x_i} \mathbf{f}_i(\mathbf{W}^{\epsilon}) = \epsilon E'(\mathbf{W}^{\epsilon}) \cdot \Delta \mathbf{W}^{\epsilon}$$

$$E'(\mathbf{W}^\epsilon)\cdot\frac{\partial}{\partial t}\mathbf{W}^\epsilon+\sum_{i=1}^{d}F_i'(\mathbf{W}^\epsilon)\cdot\frac{\partial}{\partial x_i}\mathbf{W}^\epsilon=\epsilon E'(\mathbf{W}^\epsilon)\cdot\Delta\mathbf{W}^\epsilon$$

$$\frac{\partial}{\partial t}E(\mathbf{W}^\epsilon)+\sum_{i=1}^{d}\frac{\partial}{\partial x_i}F_i(\mathbf{W}^\epsilon)=\epsilon E'(\mathbf{W}^\epsilon)\cdot\Delta\mathbf{W}^\epsilon.$$

We rewrite the right hand side of the above equation in the following way:

$$\epsilon E'(\mathbf{W}^\epsilon)\cdot\Delta\mathbf{W}^\epsilon=\epsilon\Delta E(\mathbf{W}^\epsilon)-\epsilon\sum_{i=1}^{d}\Big(\frac{\partial\mathbf{W}^\epsilon}{\partial x_i}\Big)^T E''(\mathbf{W}^\epsilon)\frac{\partial\mathbf{W}^\epsilon}{\partial x_i}$$

and since E is convex ($E''\geq 0$), we obtain the following inequality

$$\epsilon E'(\mathbf{W}^\epsilon)\cdot\Delta\mathbf{W}^\epsilon\leq\epsilon\Delta E(\mathbf{W}^\epsilon)$$

and hence, the above equation simplifies to

$$(E(\mathbf{W}^\epsilon))_t+\sum_{i=1}^{d}\frac{\partial}{\partial x_i}F_i(\mathbf{W}^\epsilon)\leq\epsilon\Delta E(\mathbf{W}^\epsilon).$$

For sufficiently smooth solutions $\mathbf{W}^\epsilon$ of (2.25), $\mathbf{W}$ can be expressed as a *vanishing viscosity solution*, i.e as the limit of $\mathbf{W}^\epsilon$ as $\epsilon\to 0$. In this case, the following theorem holds ([57]):

Theorem 2.1.6 (Vanishing viscosity limit). *Assume that for the nonlinear system of conservation laws (2.9)-(2.10) there exists an entropy pair* (E,F) *and let* $\{\mathbf{W}^\epsilon\}_{\epsilon>0}$ *a sequence of sufficiently smooth solutions of (2.25) such that*

$$\sup_{\epsilon>0}||\mathbf{W}^\epsilon||_{L^\infty(\mathbb{R}^d\times\mathbb{R}_+)^p}<C<\infty, \tag{2.26}$$

where $C>0$ *is a constant independent of* ϵ*. The limit*

$$\mathbf{W}=\lim_{\epsilon\to 0}\mathbf{W}^\epsilon\in\mathbf{L}^\infty_{loc}(\mathbb{R}^d\times\mathbb{R}_+), \tag{2.27}$$

if it exists, satisfies the entropy condition:

$$E(\mathbf{W})_t+\sum_{i=1}^{d}\frac{\partial}{\partial x_i}F_i(\mathbf{W})\leq 0 \tag{2.28}$$

in the sense of distributions on $\mathbb{R}^d\times\mathbb{R}_+$*.*

Remark 2.1.7. *[57, Remark 3.4, pp. 32] The assumption (2.26) and (2.27) are satisfied in the scalar case* ($m=1$)*. For general systems, it might be possible that only an* L^∞ *estimate on* $\mathbf{W}^\epsilon$ *is available, for example* $||\mathbf{W}^\epsilon||_\infty<C$*, which does not allow us to pass to the limit in the nonlinear term* $\mathbf{f}(\mathbf{W}^\epsilon)$*, when* $\epsilon\to 0$ *because oscillations may occur. In this case, Theorem 2.1.6 cannot be applied and some other strategies have to be used (see for example [38]). A sufficient condition, however, would be strong* L^1_{loc}*-convergence.*

Therefore, the distributional solution of (2.9)-(2.10) constructed by the method of vanishing viscosity satisfy the entropy condition (2.28) for all entropy functions E. This leads us to introduce the definition of an *entropy solution.*

Definition 2.1.8 (Entropy solution). *A weak solution* $\mathbf{W}$ *of (2.9)-(2.10) is called an entropy solution of the convex system of conservation laws (2.9)-(2.10) if* $\mathbf{W}$ *satisfies, for all convex entropy functions* E *of (2.9) and for all test functions* $\boldsymbol{\varphi} \in \mathbf{C}_0^1(\mathbb{R}^d \times \mathbb{R}_+)$, $\boldsymbol{\varphi} \geq 0$,

$$\int_0^\infty \int_{\mathbb{R}^d} E(\mathbf{W}) \cdot \boldsymbol{\varphi}_t \; d\mathbf{x} \; dt + \int_0^\infty \int_{\mathbb{R}^d} \sum_{i=1}^{d} F_i(\mathbf{W}) \cdot \boldsymbol{\varphi}_{x_i} \; d\mathbf{x} \; dt + \int_{\mathbb{R}^d} E(\mathbf{W}_0(\mathbf{x})) \cdot \boldsymbol{\varphi}(\mathbf{x}, 0) \; d\mathbf{x} \geq 0. \tag{2.29}$$

In 1D, a very important class of systems of conservation laws are the ones with the initial data consisting only of two different states and an exactly one discontinuity between them, i.e Riemann initial data; in such cases, an equivalent (much simpler in formulation) definition of the entropy condition is available due to Lax [85], who showed the existence and the stability of the entropy solutions for one-dimensional nonlinear systems of conservation laws. These conditions are called *Lax entropy conditions* (see [88] for more details).

Definition 2.1.9 (Lax entropy condition). *For a genuinely nonlinear conservation law (2.9)-(2.10) on* $\Omega = \mathbb{R}$, $\mathbf{W} \in L^\infty_{loc}(\mathbb{R} \times \mathbb{R}_+)$ *is called entropy solution of the Riemann problem*

$$\mathbf{W}_0(\mathbf{x}) = \begin{cases} \mathbf{W}_L, & \text{if} \quad \mathbf{x} \leq 0, \\ \mathbf{W}_R, & \text{if} \quad \mathbf{x} > 0, \end{cases} \tag{2.30}$$

if it satisfies (2.11) and if the jump in the k-th characteristic field is admissible,

$$\lambda_k(\mathbf{W}_L) \geq s(t) \geq \lambda_k(\mathbf{W}_R), \quad s(t) = \frac{\partial}{\partial t}\sigma(t), \tag{2.31}$$

for every $k = 1, \ldots, m$. Here, $\sigma, s : \mathbb{R}_+ \to \mathbb{R}$ denote the shock location and speed at time t.

Remark 2.1.10. *For the scalar conservation laws (i.e* $m = 1$*), the Lax entropy condition (2.30) is also known as Oleinik entropy condition [107].*

Before starting to discuss the existence and uniqueness of the entropy solution, we introduce the notion of *total variation* (TV) semi-norm $TV(w) = |w|_{TV(\Omega)}$ of $w \in L^1(\Omega)$, which is defined as the integral of the weak (distributional) derivative of w,

$$|w|_{TV(\Omega)} = \sup \Big\{ \int_\Omega w(\mathbf{x}) \operatorname{div} \boldsymbol{\varphi}(\mathbf{x}) d\mathbf{x} \mid \boldsymbol{\varphi} \in \mathbf{C}_0^1(\Omega, \mathbb{R}^d), ||\boldsymbol{\varphi}||_{L^\infty(\Omega)} \leq 1 \Big\}. \tag{2.32}$$

The corresponding space of functions with *bounded variation* (BV) is defined as:

$$BV(\Omega) = \Big\{ w \in L^1(\Omega) : ||w||_{BV(\Omega)} < \infty \Big\}, \tag{2.33}$$

where the $BV(\Omega)$-norm is combining the $L^1(\Omega)$-norm and the $TV(\Omega)$-semi-norm:

$$||w||_{BV(\Omega)} = ||w||_{L^1(\Omega)} + |w|_{TV(\Omega)}. \tag{2.34}$$

Remark 2.1.11. *Using the entropy pairs (E, F) defined in (2.22) and (2.23), Kruzkov also proved that the solution operator of the scalar conservation law is a L^1-contraction i.e*

$$\int_{\mathbb{R}^d} |w(\mathbf{x}, t) - v(\mathbf{x}, t)| d\mathbf{x} \leq \int_{\mathbb{R}^d} |w_0(\mathbf{x}) - v_0(\mathbf{x})| d\mathbf{x}, \tag{2.35}$$

for all times $t \in \mathbb{R}_+$. Here, w and v are weak entropy solution of (2.18) corresponding to initial data w_0 and v_0, respectively.

For the **scalar case** (2.18), the notion of weak entropy solutions ensures the existence and the uniqueness of the solution and the following result holds [56]:

Theorem 2.1.12. *For $w_0 \in L^\infty(\mathbb{R}^d) \cap BV(\mathbb{R}^d)$, the scalar conservation law (2.18) on $\Omega = \mathbb{R}^d$ has an entropy solution $w \in L^\infty(\mathbb{R}^d \times [0, T])$. Moreover, the nonlinear data-to-solution operator*

$$w(\cdot, t) = S(t) w_0, \ \forall t > 0 \tag{2.36}$$

satisfies the following estimates:

I. $S(t) : L^1(\mathbb{R}^d) \to L^1(\mathbb{R}^d)$ is a (contractive) Lipschitz map, i.e

$$||S(t)w_0 - S(t)u_0||_{L^1(\mathbb{R}^d)} \leq ||w_0 - u_0||_{L^1(\mathbb{R}^d)}, \ \forall w_0, u_0 \in L^1(\mathbb{R}^d), \tag{2.37}$$

thus, the entropy solutions are unique.

II. S(t) maps $(L^1 \cap BV)(\mathbb{R}^d)$ into $(L^1 \cap BV)(\mathbb{R}^d)$ and

$$TV(S(t)w_0) \leq TV(w_0), \ \forall w_0 \in (L^1 \cap BV)(\mathbb{R}^d). \tag{2.38}$$

III. For every $w_0 \in (L^1 \cap BV)(\mathbb{R}^d)$,

$$||S(t)w_0||_{L^\infty(\mathbb{R}^d)} \leq ||w_0||_{L^\infty(\mathbb{R}^d)}, \tag{2.39}$$

$$||S(t)w_0||_{L^1(\mathbb{R}^d)} \leq ||w_0||_{L^1(\mathbb{R}^d)}. \tag{2.40}$$

IV. The mapping $S(t)$ is a uniformly continuous mapping from $L^1(\mathbb{R}^d)$ into $\mathbf{C}([0, \infty); L^1(\mathbb{R}^d))$ and

$$||S(\cdot)w_0||_{\mathbf{C}([0,\infty);L^1(\mathbb{R}^d))} = \max_{0 \leq t \leq T} ||S(t)w_0||_{L^1(\mathbb{R}^d)} \leq ||w_0||_{L^1(\mathbb{R}^d)}. \tag{2.41}$$

The uniqueness is ensured based on Kružkov's result for the weak entropy solution, relying on L^1-contraction estimates (2.35). The proof is based on the *doubling of variables* idea and uses Kružkov entropies. For more details of the proof, we refer to [80]. The existence is based on the viscous approximation (2.25). For a complete proof, we refer to Godlewski and Raviat [56, Chapter II, Section 5].

For the **linear** hyperbolic system (i.e $m > 1$) of conservation laws with linear fluxes, classical existence and uniqueness results are also available (see for example LeVeque [88]).

For **nonlinear** hyperbolic system (i.e $m > 1$) of conservation laws with nonlinear fluxes, no global well-posedness results are available. The main theoretical results are available only for 1D nonlinear systems of conservation laws: for the special case of Riemann initial data, Lax showed existence and stability of entropy solutions [85]; for a general Cauchy problem, existence was obtained by Glimm [29, 55] and uniqueness and stability was studied in [25]. Glimm introduced a numerical scheme that produces approximate solutions with bounded variation. The key ingredient of this is based on Helly's theorem which states that bounded sets of BV functions have a compactness property, from where the compactness of the set of approximate solutions, and hence the existence result. For more details, we refer to Bressan [29], who extended Glimm's ideas considerably and set up a very powerful framework for studying existence and uniqueness of systems of conservation laws in 1D.

For a general Cauchy problem ($m > 1$) of type (2.9)-(2.10) in 1D, one can show the existence of a global entropy weak solution for every initial data with sufficiently small total variation (see for example [29]). Consider a domain of the form

$$\mathcal{U} = \mathrm{cl}\{\mathbf{W} \in L^1(\mathbb{R};\mathbb{R}^m); \mathbf{W} \text{ is piecewise constant}, \mathbf{V}(\mathbf{W}) + C_0\mathbf{Q}(\mathbf{W}) < \delta_0\},$$

where cl denotes the closure in L^1, the functional $\mathbf{V}(t) := \sum_\alpha |\sigma_\alpha|$ measures the total strength of waves in $\mathbf{W}(\cdot,t)$ and $\mathbf{Q}(t) := \sum_{(\alpha,\beta)} |\sigma_\alpha\sigma_\beta|$ measures the wave interaction potential. With a suitable choice of the constants C_0 and $\delta_0 > 0$, the proofs of the global existence of entropy solutions [29, Theorem 7.1, page 124] and of the global existence of front tracking approximations [29, Theorem 7.2, page 127], show that, for every initial condition $\mathbf{W}_0 \in \mathcal{U}$, one can construct a sequence of ϵ-approximate front tracking solutions converging to a weak solution $\mathbf{W}$ taking values inside $\mathcal{U}$. Since the proof of convergence relied on a compactness argument, no information was obtained on the uniqueness of the limit. The following result shows that solutions constructed by a front tracking approximations converge to a unique limit, depending Lipschitz-continuously on the initial data:

Theorem 2.1.13. *For every $\mathbf{W}_0 \in \mathcal{U}$, every sequence of ϵ-approximate solutions $\mathbf{W}_\epsilon \in \mathcal{U}$ of the Cauchy problem (2.9)-(2.10) converges to a unique limit solution $\mathbf{W} \in \mathcal{U}$ as $\epsilon \to 0$. The map $(\mathbf{W}_0, t) \mapsto \mathbf{W}(\cdot,t) := S(t)\mathbf{W}_0$ is a uniformly Lipschitz semigroup, i.e:*

$$S(0)\mathbf{W}_0 = \mathbf{W}_0, \quad S(s)(S(t)\mathbf{W}_0) = S(s+t)\mathbf{W}_0,$$

$$||S(t)\mathbf{W}_0 - S(s)\mathbf{V}_0||_{L^1(\mathbb{R}_+)} \le L\Big(||\mathbf{W}_0 - \mathbf{V}_0||_{L^1(\mathbb{R}_+)} + |t-s|\Big), \quad \textit{for all} \quad \mathbf{W}_0, \mathbf{V}_0 \in \mathcal{U}, s, t \ge 0. \tag{2.42}$$

Remark 2.1.14. *The estimate (2.42) is an equivalent form of the estimate (2.37), written for the system case of hyperbolic conservation laws.*

2.1.5 Finite volume method for scalar conservation laws

In this subsection we introduce the finite volume methods (FVM) for the solution of conservation laws and hyperbolic systems and we assume that the domain $\Omega \subseteq \mathbb{R}^d$ is bounded and Lipschitz. Finite volume formulations are based on an integral form of (2.9) and the idea is the following: instead of approximating pointwise at grid points, we split the domain in mesh/grid cells and approximate the total integral of $\mathbf{W}$ over each grid cell or over the cell average of $\mathbf{W}$, which will be divided by the volume of the cell. These values will be modified in each time step by the flux at the edges of the grid cells. The principal problem is to determine good numerical flux functions that approximate in an accurate manner the correct fluxes only based on the given data, namely, the approximate cell averages [88].

2.1.5.1 Formulation

For simplicity, we consider bounded Cartesian spatial domains, i.e, we set:

$$\Omega = I_1 \times \cdots \times I_d \subset \mathbb{R}^d,\ I_k \subset \mathbb{R},\ k = 1, \cdots d,$$

where $I_k \subset \mathbb{R}$ is bounded and connected. The discussion presented here can be extended to systems on general polyhedral domains and with suitable boundary conditions (BC).

The first step in any numerical approximation is to discretize the computational domain. Let $Q = Q^1 \times \cdots \times Q^d$ be an uniform quadrilateral mesh covering the domain Ω. The mesh consists of identical non-overlapping open cells $C_\mathbf{j}$ or also called control volumes (CV),

$$C_\mathbf{j} = C_{\mathbf{j}_1} \times \cdots C_{\mathbf{j}_d} \subset I_1 \times \cdots \times I_d \subset \mathbb{R}^d, \quad \mathbf{j}_i = 1, \cdots, \#Q^i, \quad i = 1, \cdots, d$$

and we assume for simplicity, that the mesh widths are equal in each dimension, i.e.

$$\Delta x := \frac{|I_1|}{\#Q^1} = \cdots = \frac{|I_d|}{\#Q^d}.$$

As the fluxes are defined across the cell interfaces, we denote by $\mathbf{x_j} = (x_{\mathbf{j}_1}, \cdots, x_{\mathbf{j}_d}) \in \mathbb{R}^d$ the center of each cell $C_\mathbf{j}$ and with $\mathbf{x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}$ the midpoint values between two adjacent cells midpoints $\mathbf{x_j}$ and $\mathbf{x}_{\mathbf{j}+\mathbf{e}_i}$ (in direction i):

$$\mathbf{x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i} = \frac{1}{2}(\mathbf{x_j} + \mathbf{x}_{\mathbf{j}+\mathbf{e}_i}),$$

where $\mathbf{e}_i$ denotes the set of canonical basis vectors $\{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$ of the space $\mathbb{R}^d$.

For example, in the one dimensional case, the CVs on a quadrilateral mesh are subintervals of the problem interval and the nodes can be the midpoints or the edges of the subintervals. In the 2D computational grid, the CVs are usually chosen identically with the grid cells. The nodes can be defined as the vertices or the centers of the CVs often called edge or cell-centered approaches, respectively. The CVs on a quadrilateral mesh in 3D are hexahedrons and the nodes can be chosen as the vertices of the CVs (see Figure 2.5).

We can define now the approximations to cell averages of the solution $\mathbf{W}$ as

$$\mathbf{W_j} \approx \frac{1}{|C_\mathbf{j}|} \int_{C_\mathbf{j}} \mathbf{W}(\mathbf{x}, t)\ d\mathbf{x}, \tag{2.43}$$

which are well defined, as the definition of weak solutions (2.11) only requires $\mathbf{W}$ to be integrable. We can integrate now the conservation law (2.9) over $C_\mathbf{j}$ obtaining:

$$\int_{C_\mathbf{j}} \mathbf{W}_t(\mathbf{x}, t)\ d\mathbf{x} + \sum_{i=1}^{d} \int_{C_\mathbf{j}} \frac{\partial}{\partial x_i} \mathbf{f}_i(\mathbf{W}(\mathbf{x}, t))\ d\mathbf{x} = 0$$

and using the fundamental theorem of calculus and Fubini's theorem, we obtain:

$$\int_{C_\mathbf{j}} \mathbf{W}_t(\mathbf{x}, t)\ d\mathbf{x} = -\sum_{i=1}^{d} \Big(\mathbf{f}_i(\mathbf{W}(\mathbf{x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}, t)) - \mathbf{f}_i(\mathbf{W}(\mathbf{x}_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}, t)) \Big).$$

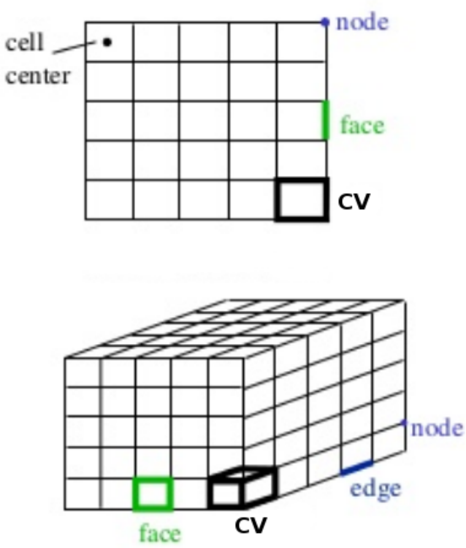


Figure 2.5: Quadrilateral mesh in two and three dimensions

The last step is to divide by $|C_{\mathbf{j}}|$, to denote the fluxes in the ith direction as

$$\mathbf{f}^{\Delta x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) := \mathbf{f}_i(\mathbf{W}(\mathbf{x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}, t)) \tag{2.44}$$

and to use (2.43), in order to obtain the semi-discrete finite volume scheme for approximating (2.9) [88]:

$$\partial_t \mathbf{W}_{\mathbf{j}}(t) = -\frac{1}{|C_{\mathbf{j}}|} \sum_{i=1}^{d} \left(\mathbf{f}^{\Delta x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) - \mathbf{f}^{\Delta x}_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}(t) \right). \tag{2.45}$$

Remark 2.1.15. *The equation (2.45) states that the rate of change of cell averages is given by the difference of the fluxes across the cell boundaries, which is exactly the definition of conservation.*

The most difficult task in the next subsection will be to approximate the fluxes $\mathbf{f}^{\Delta x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t)$.

2.1.5.2 Godunov method

Based on Godunov's idea [58], we have to approximate the fluxes in (2.45). Since we assume that for a fixed intermediate time $t_0 \geq 0$ the solution $\mathbf{W}(\mathbf{x}, t_0)$ is approximated by the cell averages $\mathbf{W}_{\mathbf{j}}(t_0)$ and hence is constant in each cell $C_{\mathbf{j}}$, for each direction $\mathbf{x}_i$ with $i \in \{1, \cdots, d\}$ and at each interface $x^i_{j+\frac{1}{2}}$, the semi-discrete formulation (2.45) is approximated (in the $\mathbf{x}_i$ direction) by a one-dimensional *Riemann problem* for $t \geq t_0$:

$$\begin{cases} \bar{\mathbf{W}}_t + \frac{\partial}{\partial \mathbf{x}_i} \mathbf{f}_i(\bar{\mathbf{W}}) & = 0, \\ \bar{\mathbf{W}}(\mathbf{x}_i, t_0) & = \begin{cases} \mathbf{W}_j(t_0), & \text{if } \mathbf{x}_i \leq x^i_{\mathbf{j}+\frac{1}{2}}, \\ \mathbf{W}_{j+\mathbf{e}_i}(t_0), & \text{if } \mathbf{x}_i > x^i_{\mathbf{j}+\frac{1}{2}}. \end{cases} \end{cases} \tag{2.46}$$

This is simply the conservation law together with a particular initial data consisting of two constant states $\mathbf{W}_j$ and $\mathbf{W}_{j+\mathbf{e}_i}$ separated by a single discontinuity at each interface $x^i_{\mathbf{j}+\frac{1}{2}}$.

Remark 2.1.16 (CFL condition). *In order to limit the time step Δt and to ensure that the waves from (2.46) are not intersecting for $t < t_0 + \Delta t$, one has to impose the Courant-Friedrichs-Lewy (CFL) condition:*

$$\max_{\mathbf{j}} \Big| \lambda_{\max}(\mathbf{x_j}, \mathbf{W_j}(t_0)) \Big| \frac{\Delta t}{\Delta x} \leq \frac{1}{2}, \tag{2.47}$$

where $\lambda_{\max}$ is the maximum eigenvalue resulting from the eigen-decomposition of all possible linear combinations of the Jacobians $D\mathbf{f}_i(\mathbf{W}) : \mathbb{R}^p \to \mathbb{R}^p$ of the flux functions $\mathbf{f}_i$, $i = 1, \dots, d$. A reformulation of this definition can be:

- *The CFL condition simply states that the method must be used in such a way that information has a chance to propagate at the correct physical speeds, as determined by the eigenvalues of the flux Jacobian $D\mathbf{f}_i(\mathbf{W})$.*

The solutions of each Riemann problem (2.46) are self-similar [88], i.e

$$\bar{\mathbf{W}}(\mathbf{x}_i, t) = \bar{\mathbf{W}}\left(\frac{\mathbf{x}_i - x^i_{j+\frac{1}{2}}}{t - t_0} \right),$$

the flux across the cell interface $\mathbf{x}_i = x^i_{j+\frac{1}{2}}$ is constant,

$$\mathbf{f}_i(\bar{\mathbf{W}}(x^i_{j+\frac{1}{2}}, t)) = \mathbf{f}_i(\bar{\mathbf{W}}(x^i_{j+\frac{1}{2}}, t_0)) =: \mathbf{f}_i(\bar{\mathbf{W}}(x^i_{j+\frac{1}{2}})),$$

where the flux $\mathbf{f}_i(\bar{\mathbf{W}}(x^i_{j+\frac{1}{2}}))$ is well defined, and we are using it to define the approximation of $\mathbf{f}^{\Delta x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t)$ in (2.45):

$$\mathbf{f}^{\Delta x}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) \approx \mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) := \mathbf{f}(\bar{\mathbf{W}}(x^i_{j+\frac{1}{2}})), \ t \in [t_0, t_0 + \Delta t). \tag{2.48}$$

Then, (2.45) together with the Riemann problem fluxes (2.48) gives the standard form of a finite volume scheme for conservation laws:

$$\partial_t \mathbf{W_j}(t) = -\frac{1}{|C_\mathbf{j}|} \sum_{i=1}^{d} \Big(\mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) - \mathbf{f}_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}(t) \Big), \tag{2.49}$$

where for scalar conservation laws ($m = 1$) and some nonlinear one-dimensional systems ($d = 1, m > 1$) of conservation laws, Riemann problems (2.46) can be solved exactly, leading to the *Godunov flux* $\mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i} = \mathbf{f}^{God}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}$ [88], which in the scalar case ($m = 1$) has the following form:

$$\mathbf{f}^{God}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(\mathbf{W}_L, \mathbf{W}_R) = \begin{cases} \min\limits_{\mathbf{W}_L \leq \xi \leq \mathbf{W}_R} \mathbf{f}_i(\xi) \text{ if } \mathbf{W}_L \leq \mathbf{W}_R, \\ \max\limits_{\mathbf{W}_R \leq \xi \leq \mathbf{W}_L} \mathbf{f}_i(\xi) \text{ if } \mathbf{W}_R > \mathbf{W}_R, \end{cases} \tag{2.50}$$

with the left and right states denoted as:

$$\mathbf{W}_L = \mathbf{W_j}(t_0), \quad \mathbf{W}_R = \mathbf{W}_{\mathbf{j}+\mathbf{e}_i}(t_0).$$

The finite volume scheme (2.49) with the Godunov flux (2.50) is called the *Godunov scheme*. Godunov's method require the solution of Riemann problems at every cell boundary (and in each time step). In theory, these Riemann problems can be solved but in practice it is computationally expensive, requiring some iterations of nonlinear equations. The structure of the Riemann solver is even not used in Godunov's method. The exact solution is averaged over each grid cell, introducing large numerical errors. Since we are approximating the exact solution $\mathbf{W}(\mathbf{x},t)$ by cell averages $\mathbf{W}_{\mathbf{j}}(t)$, this suggests that we could also estimate the fluxes $\mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}$ by solving the Riemann problem approximately. This class of finite volume schemes will be called *approximate Riemann solvers* (ARS) (not discussed here) [57, 88].

Remark 2.1.17. *In applications, one can choose from a range of various different fluxes. For example, Lax-Friedrichs flux, Rusanov flux, Roe flux etc (see [88, 131] for more details).*

The last step in the finite volume scheme (2.49) is the time integration. Firstly, we discretize the time interval, choosing the snapshots $0 < t^0 < \cdots < t^n$ with the time step $\Delta t = t^i - t^{i-1}$ and respecting the CFL condition (2.47). Then, we assemble all the cell averages $\mathbf{W}_{\mathbf{j}}$ of $\mathbf{W}$ in the collection $\{\mathbf{W}\}(t) = \{\mathbf{W}_{\mathbf{j}}(t)\}_{C_{\mathbf{j}}\in Q}$ and rewrite equation (2.49) in the operator form:

$$\frac{\partial}{\partial t}\{\mathbf{W}\}(t) = \mathcal{L}(\{\mathbf{W}\}(t)),$$

where the operator $\mathcal{L}$ acts on each cell average $\mathbf{W}_{\mathbf{j}}(t)$ of $\{\mathbf{W}\}(t)$:

$$\mathcal{L}(\mathbf{W}_{\mathbf{j}}(t)) := -\frac{1}{|C_{\mathbf{j}}|}\sum_{i=1}^{d}\Big(\mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t) - \mathbf{f}_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}(t)\Big).$$

The simplest time stepping scheme for temporal discretization of (2.49) is the *Forward Euler* (FE) method [88]:

$$\{\mathbf{W}^{n+1}\} = \{\mathbf{W}^n\} + \Delta t\mathcal{L}(\{\mathbf{W}^n\}), \tag{2.51}$$

where

$$\mathcal{L}(\{\mathbf{W}^n\}) = -\frac{1}{|C_{\mathbf{j}}|}\sum_{i=1}^{d}\Big(\mathbf{f}^n_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i} - \mathbf{f}^n_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}\Big),$$

defining

$$\mathbf{f}^n_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathbf{f}_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i}(t)dt.$$

The approximation of the solution is given in terms of cell averages $\{\mathbf{W}^n_{\mathbf{j}}\}_{C_{\mathbf{j}}\in Q}$,

$$\mathbf{W}^n_Q(\mathbf{x}) = \mathbf{W}_Q(\mathbf{x},t^n) = \mathbf{W}^n_{\mathbf{j}},\ \forall \mathbf{x}\in C_{\mathbf{j}}.$$

The method (2.51) can be also written as a direct finite differences approximation to the conservation law (2.9):

$$\frac{\{\mathbf{W}^{n+1}\} - \{\mathbf{W}^n\}}{\Delta t} + \frac{\sum_{i=1}^{d}\Big(\mathbf{f}^n_{\mathbf{j}+\frac{1}{2}\mathbf{e}_i} - \mathbf{f}^n_{\mathbf{j}-\frac{1}{2}\mathbf{e}_i}\Big)}{|C_{\mathbf{j}}|} = 0.$$

Remark 2.1.18. *In one space dimension, the finite volume method simply writes:*

$$\mathbf{W}_j^{n+1} = \mathbf{W}_j^n - \frac{\Delta t}{\Delta x}(\mathbf{f}_{j+\frac{1}{2}}^n - \mathbf{f}_{j-\frac{1}{2}}^n), \tag{2.52}$$

where Δt is the time discretization, Δx is the space discretization, $\mathbf{W}_j^n$ are the cell averages at time level n and the numerical flux $\mathbf{f}_{j+\frac{1}{2}}^n$ plays the role of an average flux through $x_{j+\frac{1}{2}}$ over the time interval $[t^n, t^{n+1}]$.

2.2 Reduced basis methods for parametrized PDEs

Reduced basis methods in particular have been applied successfully for various elliptic and parabolic problems, almost exclusively based on finite element discretizations. For linear elliptic problems we refer to [108], linear parabolic equations are treated in [62], extensions to nonlinear equations [135] or systems [121] have been developed. In [65] is proposed a RB-formulation for linear finite volume (FV) schemes in case of so called affine parameter dependence of the data functions. The latter work was extending this RB-scheme to explicit discretizations with general parameter dependence and demonstrated the applicability to a linear evolution problem [68] and to nonlinear conservation laws with explicit finite volume schemes [67]. In this context of parameterized nonlinear evolution equations, we mention the following papers, each of them proposing different methods in order to deal with the large Kolmogorov n-width, using: approximations of generalized Lax pairs [53], the solution of Monge-Kantorovich mass transfer problem [72], the method of freezing [106] or using the domain partitioning method, followed by an interpolation step [128].

2.2.1 Parametrized hyperbolic problem and the idea of reduced basis methods

A *parametrized system of m balance laws* with $m \geq 1$ on a d-dimensional (d=1,2,3) Lipschitz domain $\Omega \subseteq \mathbb{R}^d$, time domain $\mathbb{R}_+ = \{t \in \mathbb{R} : t \geq 0\}$ and with a given input parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$ is described as:

$$\begin{cases} \mathbf{W}_t(\mathbf{x}, t) + \mathcal{L}(\mathbf{x}, t; \boldsymbol{\mu})[\mathbf{W}(\mathbf{x}, t); \boldsymbol{\mu}] & = \mathbf{S}(\mathbf{x}, t, \mathbf{W}; \boldsymbol{\mu}),\ (\mathbf{x}, t) \in \Omega \times \mathbb{R}_+, \\ \mathbf{B}(\mathbf{W}; \boldsymbol{\mu}) & = \mathbf{g}(\mathbf{x}, t; \boldsymbol{\mu}),\ (\mathbf{x}, t) \in \partial\Omega \times \mathbb{R}_+, \\ \mathbf{W}(\mathbf{x}, t = 0; \boldsymbol{\mu}) & = \mathbf{W}_0(\mathbf{x}; \boldsymbol{\mu}),\ \mathbf{x} \in \Omega, \end{cases} \tag{2.53}$$

where $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$ is the space variable $(1 \leq d \leq 3)$, $t \in \mathbb{R}_+$ is the time variable, $\partial\Omega$ is the boundary of the domain, the parameter space $\mathcal{P}$ represents a closed and bounded subset of the Euclidean space $\mathbb{R}^p,\ p \geq 1$ and $\mathbf{W}_t$ denotes the time derivative of the physical variables

$$\mathbf{W} = \mathbf{W}(\mathbf{x}, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^m,\ m \geq 1,$$

which are under the conservation law if $\mathbf{S} = 0$.

The operator $\mathcal{L}(\cdot, t; \boldsymbol{\mu}) = \text{div}\ \mathbf{f}(\mathbf{W}(\cdot, t); \boldsymbol{\mu})$ represents the divergence of the nonlinear parameter dependent flux

$$\mathbf{f} : \mathbb{R}^m \times \mathcal{P} \to (\mathbb{R}^m)^d,\ \mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_d),$$

which is a collection of directional vector-valued flux functions, where

$$\mathbf{f}_i(\mathbf{W};\boldsymbol{\mu}) : \mathbb{R}^m \times \mathcal{P} \to \mathbb{R}^m,\ i = 1, \cdots, d,\ \forall \boldsymbol{\mu} \in \mathcal{P}$$

and the source term is given by:

$$\mathbf{S} : \Omega \times \mathbb{R}_+ \times \mathbb{R}^m \times \mathcal{P} \to \mathbb{R}^m.$$

To the partial differential equation are added the boundary conditions $\mathbf{g}$, which are imposed through a suitable boundary operator $\mathbf{B}$,

$$\mathbf{g} : \partial\Omega \times \mathbb{R}_+ \times \mathcal{P} \to \mathbb{R}^m$$

and the initial conditions

$$\mathbf{W}_0 : \Omega \times \mathcal{P} \to \mathbb{R}^m.$$

The vector $\boldsymbol{\mu}$ defines the system of interest and it can characterize geometric features of the computational domain, some physical or material properties of the model at hand (for example, in aerodynamics, it can represent the Mach number or the angle of attack (AoA)), initial and boundary conditions or source terms. As a result, the field variable given by the exact solution of the parametrized PDE can be seen as a *map* $\mathbf{W} : \Omega \times \mathbb{R}_+ \times \mathcal{P} \to \mathbb{R}^m$ which associates to any $\boldsymbol{\mu} \in \mathcal{P}$ a solution $\mathbf{W}(\mathbf{x}, t; \boldsymbol{\mu})$.

The discrete evolution schemes are based on approximating high-dimensional discrete space V_h (being a subset of some Hilbert space), $\dim(V_h) = N_h$, where h represents the characteristic mesh size and by approximating the exact solution at time-instances $0 = t^0 < t^1 < \dots t^K = T$ i.e providing a sequence of functions $\mathbf{W}_h^k(\boldsymbol{\mu}) : \mathbb{R}^{N_h} \to \mathbb{R}^m$ for $k = 0, \dots, K$ such that $\mathbf{W}_h^k(\boldsymbol{\mu}) \approx \mathbf{W}(t_k; \boldsymbol{\mu})$.

For any given $\boldsymbol{\mu} \in \mathcal{P}$, problem (2.53) requires to solve as many non-linear systems as the number of Newton iterations or time steps, respectively, thus involving a very high computational cost. The steps of the reduced order method (ROM) in order to fix this problem of the cost are [113]:

- Replace the discretized version of problem (2.53) with a reduced problem of dimension $N \ll N_h$ whose solution is denoted by $\mathbf{W}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ and is called *reduced basis solution* considering that
  1. the reduced problem retains the essential properties of the map $\boldsymbol{\mu} \mapsto \mathbf{W}_h(\boldsymbol{\mu})$;
  2. the error between the solution of the reduced problem $\mathbf{W}_N$ and the high-fidelity one $\mathbf{W}_h$ stays below a desired threshold. Moreover, the reliability of the reduced solution is assessed through an *a posteriori error estimator*. The estimate of the RB approximation error has to be obtained via an inexpensive (i.e., independent of the computational mesh) and rigorous (i.e., the estimation has to constitute an upper bound for the actual error) way;
  3. decoupling of the computation in two stages: an expensive *offline* stage, to be performed only once, and a very inexpensive *online* stage, in which is actually performed the *input-output* evaluation.

Hence, roughly speaking, the key idea of a reduced basis (RB) method is to generate an approximate solution to problem (2.53) belonging to a low-dimensional of dimension $N \ll N_h$ [22, 23, 69, 113].

This case of hyperbolic problems is a special, challenging one because the moving waves and discontinuities such as shocks will depend on the different parameter settings $\boldsymbol{\mu} \in \mathcal{P}$ and will develop during time. This implies that even if the structure of the problem is simple, the reduced order method requires a large number of reduced basis in order to accurately approximate these features. The task of the RB method will be to capture the evolution of both smooth and discontinuous solutions, keeping in the same time the dimension of the RB space as small as possible. Hence, the challenge is related to the fact that RB methods are based on approximating the elements of the high-fidelity solution set by a linear, global approximation under a separable form. Similar separable forms are also assumed for the fluxes or sources when they are separable into an affine decomposition. Unfortunately, if the functional $\mathcal{L}(\mathbf{x}, t; \boldsymbol{\mu})[\mathbf{W}(\mathbf{x}, t; \boldsymbol{\mu})]$ in (2.53) is not affine in the parameter, the online complexity will no longer be independent of N_h and the dimensionality reduction will not necessarily imply CPU reduction and as a result, will not admit an efficient online-offline decomposition. Hence, a further level of reduction called *hyper-reduction* will be introduced, suitably employing techniques such as EIM, which recovers online N_h independence even in the presence of non-affine parameter dependecy.

2.2.2 Reduced basis methods: basic principles and properties

As described above, the theory of RB methods for parametrized PDEs is well developed for elliptic and parabolic problems and it has been the subject of numerous studies during the last decades. For simplicity, in Section 2.2.2, we will make a brief overview of the foundation of the theory of RB methods based on the four books in this domain [22, 23, 69, 113] for parametrized linear elliptic PDEs. Then, starting with Chapter 3, we will present innovative ideas for advancing the state of art of MOR for our parametrized problem of conservation laws (2.53), taking in consideration all the difficulties described in Section 1.1 that might appear for these problems.

Parametrized variational problem

As already mentioned in Section 2.2.1, the parametrized PDEs are partial differential equations that depend on a input-parameters vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T \in \mathcal{P}$, where the input parameter set $\mathcal{P}$ is a compact subset of $\mathbb{R}^p$. We denote by $\Omega \subseteq \mathbb{R}^d, d = 1, 2, 3$ the reference domain, $V = V(\Omega)$ a suitable Hilbert space and V' its dual. We focus on the following parametrized problem written in an abstract form:

$$L(\boldsymbol{\mu}) w(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) \quad \text{in} \quad V', \tag{2.54}$$

where $\boldsymbol{\mu} \in \mathcal{P}$ is given, $L(\boldsymbol{\mu}) : V \to V'$ is a second order differential operator and $f(\boldsymbol{\mu}) : V \to \mathbb{R}$ denotes a linear and continuous form on V, that is an element of V'. In a weak form, the abstract formulation (2.54) can be seen as:

$$a(w(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in V \tag{2.55}$$

where $a : V \times V \times \mathcal{P} \to \mathbb{R}$ is obtained from $L(\boldsymbol{\mu})$ and is written in a bilinear form, where the bilinearity is with respect to the first two variables and $f : V \times \mathcal{P} \to \mathbb{R}$ is written in a linear form, where the linearity is with respect to the first variable.

In order to state a well-posed problem for all parameters values $\boldsymbol{\mu} \in \mathcal{P}$, we assume in addition to the bilinearity and the linearity of the parametrized forms $a(\cdot, \cdot; \boldsymbol{\mu})$ and $f(\cdot; \boldsymbol{\mu})$ that

- $a(\cdot, \cdot; \boldsymbol{\mu})$ is *continuous and stable* over $V \times V$ for all $\boldsymbol{\mu} \in \mathcal{P}$ with respect to the norm $|| \cdot ||_V = (v, v)_V^{1/2}$ induced by the inner product $(v, v)_V^{1/2}$ defined over V i.e there exists a finite constant $\gamma(\boldsymbol{\mu}) \leq \gamma < \infty, \gamma > 0$ called *continuity factor* of $a(\cdot, \cdot, \boldsymbol{\mu})$ such that

$$\gamma(\boldsymbol{\mu}) = \sup_{v \in V} \sup_{w \in V} \frac{a(v, w; \boldsymbol{\mu})}{||v||_V ||w||_V} < \gamma, \quad \forall \boldsymbol{\mu} \in \mathcal{P}$$

  and a positive constant $\beta(\boldsymbol{\mu}) \geq \beta > 0$ called *inf-sup stability factor* such that

$$\beta(\boldsymbol{\mu}) = \inf_{v \in V} \sup_{w \in V} \frac{a(v, w; \boldsymbol{\mu})}{||v||_V ||w||_V} \geq \beta, \quad \forall \boldsymbol{\mu} \in \mathcal{P}.$$

  Remark 2.2.1. *In particular, if there exists a positive constant $\alpha(\boldsymbol{\mu}) \geq \alpha > 0$ defined as*

$$\alpha(\boldsymbol{\mu}) = \inf_{v \in V} \frac{a(v, v; \boldsymbol{\mu})}{||v||_V^2} \geq \alpha, \quad \forall \boldsymbol{\mu} \in \mathcal{P} \tag{2.56}$$

  which satisfies the inf-sup stability factor, then $a(\cdot, \cdot; \boldsymbol{\mu})$ is coercive and $\alpha(\boldsymbol{\mu})$ is called coercivity factor.

- $f(\cdot; \boldsymbol{\mu})$ is *continuous* for all $\boldsymbol{\mu} \in \mathcal{P}$ with respect to the norm $|| \cdot ||_V$ i.e there exists a constant $\delta(\boldsymbol{\mu}) \leq \delta < \infty, \delta > 0$ called *continuity factor* of $f(\cdot; \boldsymbol{\mu})$ such that

$$\delta(\boldsymbol{\mu}) = \sup_{v \in V} \frac{f(v; \boldsymbol{\mu})}{||v||_V} < \delta, \quad \forall \boldsymbol{\mu} \in \mathcal{P}$$

Thanks to the continuity and stability properties, it is clear that (2.55) admits a unique solution, based on Nečas theorem who proved that weakly coercive problems are well posed [104]. Moreover, the following stability estimate holds for all $\boldsymbol{\mu} \in \mathcal{P}$:

$$||w(\boldsymbol{\mu})||_V \leq \frac{1}{\beta(\boldsymbol{\mu})} ||f(\cdot; \boldsymbol{\mu})||_{V'} \leq \frac{1}{\beta} ||f(\cdot; \boldsymbol{\mu})||_{V'}.$$

Discretization Techniques

Consider now a discrete approximation of the weak formulation (2.55) i.e there is a discrete approximation space $V_h \subset V$, $\dim V_h = N_h$ in which the approximate solution is sought. For example, the approximation space V_h can be constructed as a standard finite element method based on a triangulation and using piecewise linear basis functions. This finite dimensional space V_h inherits the norm $|| \cdot ||_V$ from the Hilbert space V, unless otherwise stated. In the discretized form, problem (2.55) writes:

$$a(w_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}), \quad \forall v_h \in V_h, \tag{2.57}$$

which can be equivalently written as

$$L(\boldsymbol{\mu})w_h(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) \quad \text{in} \quad V_h^{'}. \tag{2.58}$$

We say that this formulation is called the *truth problem* or *the high-fidelity model* and its solution computed for only one parameter is called the *truth approximation* and it can be achieved with as high accuracy as desired. However, the computational cost of the truth solution is extremely expensive since the space V_h may involve many *degrees of freedom* (DOF) N_h to achieve the desired accuracy level. The Gallerkin high-fidelity approximation (2.57) can be written in the following linear system formulation:

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{w}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}), \tag{2.59}$$

where $\{\rho^i\}_{i=1}^{N_h}$ denotes a basis for V_h, $\mathbf{A}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the parameter dependent stiffness matrix and $\mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the parameter dependent right-hand side vector and their components are

$$(\mathbf{A}_h(\boldsymbol{\mu}))_{ij} = a(\rho^i, \rho^j; \boldsymbol{\mu}), \quad (\mathbf{f}_h(\boldsymbol{\mu}))_i = f(\rho^i; \boldsymbol{\mu}), \quad 1 \leq i, j \leq N_h.$$

2.2.2.1 The solution manifold and the reduced basis approximation

Solving the high-fidelity problem (2.57) even for only one parameter $\boldsymbol{\mu} \in \mathcal{P}$ leads to a very high computational cost. Not to mention, that these kind of parametrized problems require repetitive evaluations for different parameters $\boldsymbol{\mu} \in \mathcal{P}$. These costs can be reduced by only using a suitable reduced order approximation as an alternative to solving the truth problem several times.

We start by introducing the *solution manifold* of the high-fidelity solutions $w_h(\boldsymbol{\mu})$ generated as $\boldsymbol{\mu}$ varies in the parameter domain $\mathcal{P}$:

$$\mathcal{M}_h = \{w_h(\boldsymbol{\mu}) \in V_h : \boldsymbol{\mu} \in \mathcal{P}\} \subset V_h. \tag{2.60}$$

The final goal of the reduced basis methods, as we mentioned in Section 2.2.1, is to approximate any member of the solution manifold $\mathcal{M}_h$ with a low number of, let's say N, reduced basis functions. Based on [113], in order to set up a RB method, one has to follow the next steps:

1. **Reduced basis construction.** Generate a set of N *reduced basis functions* $\{\zeta_1, \ldots, \zeta_N\}$ by orthonormalizing the elements of a high-fidelity solutions set $\{w_h(\boldsymbol{\mu}^i)\}_{i=1}^N$ with respect to a suitable scalar product, elements which are also called *snapshots* and which corresponding to a set of N selected parameters $\{\boldsymbol{\mu}^i\}_{i=1}^N$. Then, the *reduced basis space* is defined as

$$V_N = \text{span}\{\zeta_1, \ldots, \zeta_N\} = \text{span}\{w_h(\boldsymbol{\mu}^1), \ldots, w_h(\boldsymbol{\mu}^N)\} \subset V_h. \tag{2.61}$$

The spaces $\{V_N, N \geq 1\}$ are nested, that is $V_{N-1} \subset V_N, N \geq 2$. This hierarchical choice of the spaces is not necessary. Nevertheless, it turns out to be very useful because it allows a better exploitation of the memory during the computation and, as a consequence, this improves the efficiency of the method.

Remark 2.2.2. *The reduced basis construction will be based on a greedy algorithm, where the snapshots are selected according to a suitable criteria. We will define this iterative sampling method and also other approaches in Section 2.2.2.4.*

2. **Reduced solution.** Since the reduced basis functions are given by $\{\zeta_1, \dots, \zeta_N\}$, we can represent the *reduced basis solution* $w_N(\boldsymbol{\mu}) \in V_N$ as a linear combination of the reduced basis functions, namely:

$$w_N(\boldsymbol{\mu}) = \sum_{i=1}^{N} w_N^{(i)}(\boldsymbol{\mu})\zeta_i, \tag{2.62}$$

where $\mathbf{w}_N(\boldsymbol{\mu}) = (w_N^{(1)}(\boldsymbol{\mu}), \dots, w_N^{(N)}(\boldsymbol{\mu})) \in \mathbb{R}^N$ represents the unknown RB coefficients.

3. **Determination of the unknown coefficients.** The coefficients $\{(w_N^{(i)}(\boldsymbol{\mu}))_{i=1}^N\}$ are determined by requiring that a suitable geometric orthogonality criteria to be fulfilled. The most famous method is the Galerkin projection-based method and it will be presented in Section 2.2.2.2.

Figure 2.6 represents a visualization of how we can approximate the truth solution $w_h(\boldsymbol{\mu})$ with a reduced one $w_N(\boldsymbol{\mu})$ by only performing a projection onto a low-dimensional V_N.

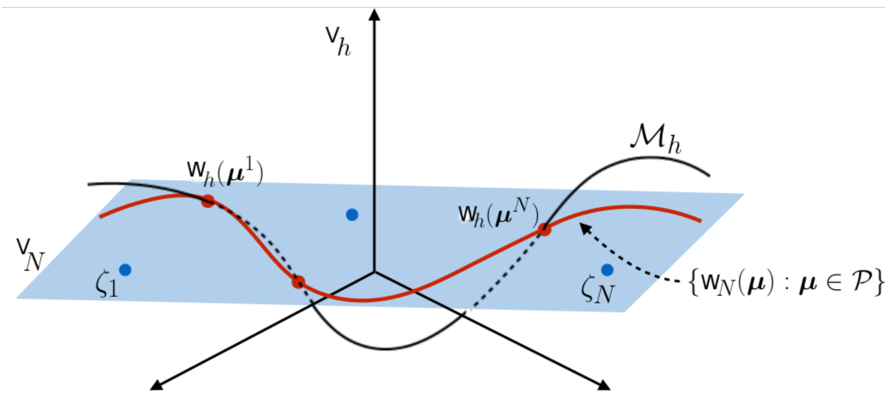


Figure 2.6: Intuitive representation of the truth manifold $\mathcal{M}_h$ (black line) and its RB approximation (red line) in the case of $p = 1$.

2.2.2.2 Galerkin projection

We focus now on steps 2 and 3 from Section 2.2.2.1. As anticipated, we generate the reduced problem via a projection approach. More precisely, the reduced problem will consist of a set of N equations that are obtained by imposing N (independent) conditions. Recalling that $N \ll N_h$, this means, that to find the RB solution we need just to solve a $N \times N$ linear system, instead of $N_h \times N_h$. Now, we can enforce the orthogonality of the residual of the high-fidelity problem (2.58) computed on the RB solution to the functions of a subspace $W_N \subset V_h$ (also called *test subspace*):

$$r(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) - L(\boldsymbol{\mu})w_N(\boldsymbol{\mu}). \tag{2.63}$$

This yields to the following *Petrov-Galerkin reduced basis* (PG-RB) problem [22]

$$\begin{aligned} < L(\boldsymbol{\mu})w_N(\boldsymbol{\mu}) - f(\boldsymbol{\mu}), w_N >= 0 \quad \forall w_N \in W_N \quad \text{or} \\ a(w_N(\boldsymbol{\mu}), w_N; \boldsymbol{\mu}) = f(w_N; \boldsymbol{\mu}) \quad \forall w_N \in W_N. \end{aligned} \tag{2.64}$$

If the test subspace W_N coincides with V_N, then (2.64) corresponds to the *Galerkin reduced basis* (G-RB) problem [22, 113].

Given $\boldsymbol{\mu} \in \mathcal{P}$, the Galerkin reduced basis approximation of problem (2.55) reads:

$$a(w_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}), \quad \forall v_N \in V_N. \tag{2.65}$$

In the case when $a(\cdot, \cdot; \boldsymbol{\mu})$ is coercive (i.e fulfills (2.56)) for any $\boldsymbol{\mu} \in \mathcal{P}$, one can prove by only applying the Lax-Milgram theorem that the well-posedness of the G-RB problem is therefore inherited from the one of the high-fidelity problem. That is, if $a(\cdot, \cdot; \boldsymbol{\mu})$ is coercive for any $\boldsymbol{\mu} \in \mathcal{P}$ over $V_h \times V_h$, then it is coercive also over $V_N \times V_N$.

We denote by $(v, u)_{\boldsymbol{\mu}} = a(v, u; \boldsymbol{\mu})$ and $||v||_{\boldsymbol{\mu}} = \sqrt{(v, v)_{\boldsymbol{\mu}}}, \forall v, u \in V$ the inner product and the energy norm induced by the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$, provided that it is symmetric for any $\boldsymbol{\mu} \in \mathcal{P}$. Subtracting (2.65) from (2.57) we obtain:

$$a(w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = 0, \quad \forall v_N \in V_N, \tag{2.66}$$

which in the symmetric coercive case, this is a *Galerkin orthogonality property* for the reduced problem, as it expresses the orthogonality of the error $w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})$ to the subspace V_N, according to the scalar product $(\cdot, \cdot)_{\boldsymbol{\mu}}$. The RB solution $w_N(\boldsymbol{\mu})$ is therefore the projection of $w_h(\boldsymbol{\mu})$ onto V_N, according to the scalar product $(\cdot, \cdot)_{\boldsymbol{\mu}}$. This proves why the G-RB method is a projection-based method and the following property is satisfied [113].

Proposition 2.2.3. *If $a(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric and coercive, then the solution $w_N(\boldsymbol{\mu}) \in V_N$ to (2.65) satisfies the following optimality property*

$$w_N(\boldsymbol{\mu}) = \arg \min_{v \in V_N} ||w_h(\boldsymbol{\mu}) - v||^2_{\boldsymbol{\mu}}. \tag{2.67}$$

In the case when $a(\cdot, \cdot; \boldsymbol{\mu})$ is not coercive over $V_h \times V_h$, the well-posedness of the G-RB problem can be proved using Babuška theorem for the case when the trial space and the test space coincide.

Introducing (2.62) in (2.65) and choosing $v_N = \zeta_n, 1 \leq n \leq N$, we obtain a set of N linear algebraic equations

$$\begin{aligned} \sum_{i=1}^{N} a(\zeta_i, \zeta_n; \boldsymbol{\mu}) w_N^{(i)}(\boldsymbol{\mu}) = f(\zeta_n; \boldsymbol{\mu}), \quad 1 \leq n \leq N, \text{or} \\ \mathbf{A}_N(\boldsymbol{\mu})\mathbf{w}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}), \end{aligned} \tag{2.68}$$

thus, we obtain a $N \times N$ linear system where the matrix $\mathbf{A}_N(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ has the elements $(\mathbf{A}_N(\boldsymbol{\mu}))_{nm} = a(\zeta_m, \zeta_n; \boldsymbol{\mu})$ and the vector $\mathbf{f}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ has the components $(\mathbf{f}_N(\boldsymbol{\mu}))_n = f(\zeta_n; \boldsymbol{\mu})$.

Remark 2.2.4. *From the computational point of view, the system (2.68) is much faster and less expensive to solve than the original high-fidelity system. Anyway, the assembly of the reduced matrix* $\mathbf{A}_N(\boldsymbol{\mu})$ *and the vector* $\mathbf{f}_N(\boldsymbol{\mu})$ *still involves computations whose complexity depends on* N_h*.*

In order to fix the problem described in Remark 2.2.4, we are using the affine decomposition with respect to the parameter $\boldsymbol{\mu}$ of the bilinear form a and of the linear form f, i.e

$$\begin{aligned} a(w,v;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) a_q(w,v) \quad \forall v,w \in V, \boldsymbol{\mu} \in \mathcal{P} \\ f(v;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f_q(v) \quad \forall v \in V, \boldsymbol{\mu} \in \mathcal{P}, \end{aligned} \tag{2.69}$$

where $\theta_a^q : \mathcal{P} \to \mathbb{R}, 1 \leq q \leq Q_a$ and $\theta_f^q : \mathcal{P} \to \mathbb{R}, 1 \leq q \leq Q_f$ are the $\boldsymbol{\mu}$-dependent functions and $a_q : V \times V \to \mathbb{R}, f_q : V \to \mathbb{R}$ are the parameter independent forms. This yields to the following expression of the RB matrix $\mathbf{A}_N$ and vector $\mathbf{f}_N$

$$\mathbf{A}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q \quad \text{and} \quad \mathbf{f}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) \mathbf{f}_N^q, \tag{2.70}$$

where the parameter independent matrices $\mathbf{A}_N^q$ and vectors $\mathbf{f}_N^q$ are given by

$$(\mathbf{A}_N^q)_{nm} = a_q(\zeta_m, \zeta_n), \quad (\mathbf{f}_N^q)_m = f_q(\zeta_m), \quad 1 \leq m, n \leq N.$$

So the systems in (2.68) can be rewritten as:

$$\Big(\sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q \Big) \mathbf{w}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) \mathbf{f}_N^q. \tag{2.71}$$

In order to compute the matrices $\mathbf{A}_N^q$ and $\mathbf{f}_N^q$ we recall that the basis function $\zeta_m \in \mathbb{V}_h, 1 \leq m \leq N$ and thus, the RB matrices and vectors can be computed from the corresponding high-fidelity ones. Indeed, expanding each RB basis function with respect to the basis functions $\{\rho^i\}_{i=1}^{N_h}$ of V_h, we obtain:

$$\zeta_m = \sum_{i=1}^{N_h} \zeta_m^{(i)} \rho^i, \quad 1 \leq m \leq N \tag{2.72}$$

and denoting by $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ the matrix whose columns are the coefficients of the RB functions in (2.72) it follows that

$$\begin{aligned} \mathbf{A}_N^q &= \mathbf{V}^T \mathbf{A}_h^q \mathbf{V}, \quad 1 \leq q \leq Q_a \\ \mathbf{f}_N^q &= \mathbf{V}^T \mathbf{f}_h^q, \quad 1 \leq q \leq Q_f \end{aligned}$$

where $(\mathbf{A}_h^q)_{ij} = a_q(\rho^j, \rho^i)$ and $(\mathbf{f}_h^q)_i = f_q(\rho^i)$ for $1 \leq i, j \leq N$.

Offline-online decomposition

Making use of the advantage of the affine decomposition, we can split the assembly of the reduced matrices and vectors in two different phases *offline-online decomposition*. In this approach, the complexity of the offline stage depends on the complexity of the approximation of the PDE, while the complexity of the online stage depends solely on the complexity of the reduced order model. When combined with a posteriori error estimates, the online stage guarantees the accuracy of the high-fidelity approximation at the low cost of a reduced order model.

It is crucial to note that in (2.71) the matrices $\mathbf{A}_N^q$ and $\mathbf{f}_N^q$ do not depend on the parameter $\boldsymbol{\mu}$. So, a good computational strategy is to compute and store them once for all. The computation and storage of the $\boldsymbol{\mu}$-independent structures is called "Offline" stage. More precisely in this stage we compute and store:

- The matrices $\mathbf{A}_h^q$, for $q = 1, \ldots, Q_a$ and the right hand side terms $\mathbf{f}_h^q$ for $q = 1, \ldots, Q_f$;
- The snapshot solutions and the corresponding orthonormal basis $\{\zeta_n\}_{n=1}^N$;
- The RB matrices $\mathbf{A}_N^q$, for $q = 1, \ldots, Q_a$ and the right hand side terms $\mathbf{f}_N^q$ for $q = 1, \ldots, Q_f$.

We recall that our aim is to obtain, given a new value $\boldsymbol{\mu} \in \mathcal{P}$, a fast and reliable approximation of $w_h(\boldsymbol{\mu})$. To do this, we need to evaluate the coefficients θ_a^q and θ_f^q in order to assemble the $N \times N$ system in (2.71). Once this system has been solved, the RB solution is obtained trough the relation (2.68). The operations done to perform the evaluation $\boldsymbol{\mu} \mapsto w_N(\boldsymbol{\mu})$ constitute the "Online" stage.

2.2.2.3 Kolmogorov n-width

Concerning the approximability of the solution set, we start asking how well $\mathcal{M}_h$ can be approximated (uniformly with respect to $\boldsymbol{\mu}$) by a finite-dimensional subspace of prescribed dimension. To answer this question, we refer to the *Kolmogorov n-width* [100, 109].

Let K be a compact set of a generic Hilbert space X, and consider a generic n-dimensional subspace $X_n \subset X$. If we define the distance between an element $x \in X$ and X_n as

$$d(x; X_n) = \inf_{x_n \in X_n} ||x - x_n||_X \tag{2.73}$$

any element $\hat{x}_n \in X_n$ which realizes the infimum, that is

$$||x - \hat{x}_n||_X = d(x; X_n), \tag{2.74}$$

is called the best approximation of x in X_n. A very natural question is whether the n-dimensional subspace is suitable to approximate all the elements $x \in K$.

To be precise, we quantify the worst possible best approximation as the angle between the subspace X_n and the set K defined by

$$d(K; X_n) = \sup_{x \in K} d(x; X_n). \tag{2.75}$$

The distance between a subspace X_n and K is determined by the worst-case scenario. Finding the best n-dimensional subspace of X for approximating K determines the minimum, over all possible n-dimensional subspaces of X, of the deviation 2.75, that is,

$$d_n(K;X) = \inf_{\substack{X_n \subset X \\ \dim(X_n)=n}} d(K;X_n) = \inf_{\substack{X_n \subset X \\ \dim(V_n)=n}} \sup_{x \in K} \inf_{x_n \in X_n} ||x - x_n||_V. \tag{2.76}$$

The number $d_n(K;X)$ is called the *Kolmogorov n-width* of K, first introduced by Kolmogorov [79]. It represents the best achievable accuracy in the V-norm when all possible elements of K are approximated by elements belonging to a linear n-dimensional subspace $X_n \subset X$. A subspace X_n of dimension at most n such that

$$d(K;\hat{X}_n) = d_n(K;X) \tag{2.77}$$

is called an *optimal n-dimensional subspace* for $d_n(K;X)$ [113].

Replacing X by V_h and K by $\mathcal{M}_h$, we can now define the Kolmogorov n-width of the solution set $\mathcal{M}_h$ as

$$d_n(\mathcal{M}_h;V_h) = \inf_{\substack{V_n \subset V_h \\ \dim(V_n)=n}} d(\mathcal{M}_h;V_n) = \inf_{\substack{V_n \subset V \\ \dim(V_n)=n}} \sup_{\boldsymbol{\mu} \in \mathcal{P}} \inf_{v_n \in V_n} ||u_h(\boldsymbol{\mu}) - v_n||_V. \tag{2.78}$$

Since V_h is a Hilbert space, there exists an orthogonal projection operator $\Pi_{V_n} : V \to V_n$ such that

$$||v - \Pi_{V_n} v||_V = \min_{v_n \in V_n} ||v - v_n||_V, \quad \forall v \in V_h. \tag{2.79}$$

The Kolmogorov n-width of $\mathcal{M}_h$ can thus be expressed as

$$d_n(\mathcal{M}_h;V_h) = \inf_{\substack{V_n \subset V_h \\ \dim(V_n)=n}} ||u_h - \Pi_{V_n} u_h||_{L^\infty(\mathcal{P};V)}. \tag{2.80}$$

For $n = N$, 2.76 corresponds to the best achivable error in a uniform sense when approximating the solution manifold $\mathcal{M}_h$ by elements of the RB space V_N. In this regard, the Kolmogorov n-width is relevant for deciding whether or not a given parametrized problem can be efficiently reduced [113].

2.2.2.4 Basis generation

This chapter corresponds to step one in Section 2.2.2.1 and in here, we will present the two most used methods for generating the reduced basis spaces: *proper orthogonal decomposition* (POD) and the *greedy sampling algorithm.*

Proper orthogonal decomposition

POD is a technique for reducing the dimensionality of a given parameter space, in which one samples this dataset, compute the truth solutions at all sample points and, following a compression step, retains only the essential information. In the theory of stochastic processes this procedure is also knows as *Karhunen-Loève (KL) decomposition* and in multivariate statistics it is precisely the *principal component analysis* (PCA). POD was firstly applied in the context of turbulent flows [127]. In the context of ROM, was used to build reduced-order models of time-dependent problems [82] but there are many applications also in the context of parametrized systems [31, 32].

Consider a set $E_s = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{n_s}\}$ of n_s parameter samples and the corresponding set of snapshots $\{w_h(\boldsymbol{\mu}^1), \dots, w_h(\boldsymbol{\mu}^{n_s})\}$, which are the solutions of the high-fidelity problem (2.57). We define the snapshot matrix $\mathbb{S} \in \mathbb{R}^{N_h \times n_s}$ as $\mathbb{S} = [\mathbf{w}_1| \dots |\mathbf{w}_{n_s}]$, where the vectors $\mathbf{w}_i \in \mathbb{R}^{N_h}, 1 \le i \le n_s$, represent the degrees of freedom of the functions $w_h(\boldsymbol{\mu}^i) \in V_h$ (i.e $\mathbf{w}_i^{(j)} = w_h^{(j)}(\boldsymbol{\mu}^i)$ for $1 \le i \le n_s$ and $1 \le j \le N_h$). Applying now the singular value decomposition of $\mathbb{S}$, we obtain:

$$\mathbb{S} = \mathbb{U}\Sigma\mathbb{Z}^T, \tag{2.81}$$

where $\mathbb{U} = [\boldsymbol{\zeta}_1| \dots |\boldsymbol{\zeta}_{N_h}] \in \mathbb{R}^{N_h \times N_h}$ and $\mathbb{Z} = [\boldsymbol{\psi}_1| \dots |\boldsymbol{\psi}_{n_s}] \in \mathbb{R}^{n_s \times n_s}$ are orthogonal matrices, and $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{N_h \times n_s}$ with $\sigma_1 \ge \sigma_2 \ge \dots \ge \sigma_r$. Here $r \le \min(N_h, n_s)$ denotes the rank of $\mathbb{S}$, which is strictly smaller than n_s if the snapshot vectors are not all linearly independent. Then, we can write

$$\mathbb{S}^T\mathbb{S}\boldsymbol{\psi}_i = \sigma_i^2\boldsymbol{\psi}_i \quad \text{and} \quad \mathbb{S}\mathbb{S}^T\boldsymbol{\zeta}_i = \sigma_i^2\boldsymbol{\zeta}_i, \quad i = 1, \dots, r \tag{2.82}$$

i.e $\sigma_i^2, i = 1, \dots, r$ represents the nonzero eigenvalues of the matrix $\mathbb{S}^T\mathbb{S}$ listed in nondecreasing order. The matrix $\mathbb{C} = \mathbb{S}^T\mathbb{S} \in \mathbb{R}^{n_s \times n_s}$ is called *correlation matrix* and its elements are given by

$$\mathbb{C}_{ij} = \mathbf{w}_i^T\mathbf{w}_j, \quad 1 \le i, j \le n_s. \tag{2.83}$$

For any $N \le n_s$, the *POD basis* $\mathbb{V} \in R^{N_h \times N}$ of dimension N is defined as the set of the first N left singular vectors of $\mathbb{U}$.

By construction, the POD basis is orthonormal and it minimizes, over all possible N-dimensional orthonormal bases $\mathbb{W} = [\mathbf{w}_1| \dots |\mathbf{w}_N] \in \mathbb{R}^{N_h \times N}$, the sum of the squares of the errors between each snapshot vector $\mathbf{w}_i$ and its projection onto the subspace spanned by $\mathbb{W}$.

Proposition 2.2.5. *Let $\mathcal{V}_N = \{\mathbb{W} \in \mathbb{R}^{N_h \times N} : \mathbb{W}^T\mathbb{W} = \mathbb{I}_N\}$ be the set of all N-dimensional orthonormal bases. Then,*

$$\sum_{i=1}^{n_s} ||\mathbf{w}_i - \mathbb{V}\mathbb{V}^T\mathbf{w}_i||_2^2 = \min_{\mathbb{W} \in \mathcal{V}_N} \sum_{i=1}^{n_s} ||\mathbf{w}_i - \mathbb{W}\mathbb{W}^T\mathbf{w}_i||_2^2 = \sum_{i=N+1}^{r} \sigma_i^2.$$

Based on the previous proposition, it follows that the error in the POD basis is equal to the sum of the squares of the singular values corresponding to the neglected POD modes. This result directly suggests a suitable criteria to select the minimal POD dimension $N \le r$ such that the projection error is smaller than a desired tolerance ϵ_{POD}. Indeed, then is sufficient to choose N as the smallest integer such that

$$I(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \ge 1 - \epsilon_{POD}^2 \tag{2.84}$$

i.e, the energy retained by the last $r - N$ modes is equal or smaller than ϵ_{POD}^2. $I(N)$ represents the percentage of energy of the snapshots captured by the first N POD modes [10].

Greedy sampling algorithm

Another very popular method to construct the reduced basis is the so called greedy algorithm [111, 112], which is an iterative algorithm where at each iteration one new basis function is added and the precision of the basis set is improved. It only requires one truth solution to be computed per iteration and a total of N truth solutions to generate the N-dimensional reduced basis space V_N in comparison with the POD method which might entail a severe computational cost due to the large number n_s of snapshots of the high-fidelity problem. As a result, greedy algorithms allow the construction of the reduced space by minimizing the amount of snapshots to be evaluated. The goal of such a procedure is to evaluate N snapshots to construct a RB space of dimension N, by seeking at each step the local optimum. An essential aspect of the greedy algorithm is the availability of an a posteriori error estimate for the error $||w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})||_V$, whose evaluation must be performed in a very inexpensive way for any $\boldsymbol{\mu} \in \mathcal{P}$.

In order to perform a greedy procedure, let us define the train samples set $\mathbb{E}_{train}$ as a finite subset of $\mathcal{P}$, with cardinality $|\mathbb{E}_{train}| = n_{train}$. Such a training sample serves to select our RB space and we need n_{train} to be large enough to ensure that $\mathbb{E}_{train}$ is a good "approximation" of the parameter space $\mathcal{P}$, i.e even if we would refine the parameter samples, the greedy algorithm should return the same results [69, 122]. We assume here that an a posteriori error estimate Δ_n is available for any $\boldsymbol{\mu} \in \mathcal{P}$ such that at each step $n = 1, \dots, N-1$

$$||w_h(\boldsymbol{\mu}) - \mathbb{V}w_N(\boldsymbol{\mu})||_{\mathbb{X}_h} \leq \Delta_n(\boldsymbol{\mu}), \forall \boldsymbol{\mu} \in \mathcal{P}, \tag{2.85}$$

where $\mathbb{X}_h$ is a symmetric positive definite matrix associated to the scalar product in V, i.e $(\mathbb{X}_h)_{ij} = (\rho^i, \rho^j)_V$. The algorithm is recursive and each step $n = 1, \dots, N-1$ is composed of two sub-steps:

- Evaluate the a posteriori error bound $\Delta_n(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathbb{E}_{train}$
- Solve the following problem

$$\boldsymbol{\mu}^{n+1} = \arg \max_{\boldsymbol{\mu} \in \mathbb{E}_{train}} \Delta_n(\boldsymbol{\mu}),$$

  i.e at the n-th iteration of this algorithm to the retained snapshots, over all possible candidates $w_h(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathbb{E}_{train}$, we append the particular candidate snapshot that the a posteriori error bound predicts to be the worst approximated by the RB prediction associated to V_n. Then, the final size N of the RB space V_N is such that

$$\max_{\boldsymbol{\mu} \in \mathbb{E}_{train}} \Delta_N(\boldsymbol{\mu}) \leq \epsilon,$$

  where ϵ is a prescribed, sufficiently small, stopping tolerance.

Remark 2.2.6.
- *The basis $\mathbb{V}$ is kept orthonormal by iteratively orthonormalizing the new element appended to the existing basis through a Gram-Schmidt procedure.*
- *Some other estimators can be used in order to evaluate the accuracy of the RB space. However, its evaluation must be not expensive.*
- *In cases where we do not have sufficient information, the train samples can be chosen by using Monte Carlo methods with respect to a uniform or a log-uniform density.*

A posteriori error estimation

One of the most important features of the reduced basis method is the *a posteriori error estimation.* As we have seen above, the estimators Δ_N play a crucial role in the construction of the RB space. For our purposes, a good a posteriori error estimator have be:

- *Rigurous:* The inequality $||w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})||_{V_h} \leq \Delta_N(\boldsymbol{\mu})$ must hold for all $\boldsymbol{\mu} \in \mathcal{P}$. This is a fundamental requirement to ensure reliability to the RB method.
- *Sharp:* It should be as close as possible to the actual (unknown) error.
- *Computationally efficient:* The computation of the error bound must be very inexpensive both to speed up the Offline stage (i.e. greedy algorithm) and to allow its use in the Online stage. The computational cost should be independent of N_h.

A posteriori error estimators are computable indicators which employ the residual of the approximate RB solution to derive estimates of the actual solution error. As a result, establishing an error-residual relationship is crucial to derive a posteriori error estimates. We observe that the error between the high-fidelity and reduced solutions $e_h(\boldsymbol{\mu}) = w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu}) \in V_h$ satisfies from (2.57) and (2.65)

$$a(e(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = a(w_h(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) - a(w_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h. \tag{2.86}$$

Then, we can define the residual $r(\cdot; \boldsymbol{\mu}) \in V_h'$ of the high-fidelity problem computed on the RB solution, introduced in (2.63) as

$$r(v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) - a(w_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h. \tag{2.87}$$

Using the continuity and the stability properties defined in the beginning of Section 2.2.2, we obtain the following estimate:

$$\frac{1}{\gamma_h(\boldsymbol{\mu})}||r(\cdot; \boldsymbol{\mu})||_{V_h'} \leq ||e_h(\boldsymbol{\mu})||_V \leq \frac{1}{\beta_h(\boldsymbol{\mu})}||r(\cdot; \boldsymbol{\mu})||_{V_h'} \tag{2.88}$$

which means that the norm of the error is bounded from below and from above by the dual norm of the residual. Since $r(\cdot; \boldsymbol{\mu})$ only involves the high-fidelity arrays and the computed reduced solution $w_N(\boldsymbol{\mu})$ but not $w_h(\boldsymbol{\mu})$, its norm can serve as an a posteriori error estimator. Then, from (2.88) we denote by

$$\Delta_N(\boldsymbol{\mu}) = \frac{||r(\cdot; \boldsymbol{\mu})||_{V_h'}}{\beta_h(\boldsymbol{\mu})} \tag{2.89}$$

the *error estimator* and its associated *effectivity factor* by

$$\eta_N(\boldsymbol{\mu}) = \frac{\Delta_N(\boldsymbol{\mu})}{||e_h(\boldsymbol{\mu})||_V}, \tag{2.90}$$

which measures the quality of the proposed estimator and for sharpness, it is considered to be close to one. Hence, we can define an equivalent version of the error bounds defined in (2.88) by

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq \frac{\gamma_h(\boldsymbol{\mu})}{\beta_h(\boldsymbol{\mu})}, \quad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{2.91}$$

Since the stability factors $\beta_h(\boldsymbol{\mu})$ and $\gamma_h(\boldsymbol{\mu})$ are the minimum and the maximum singular values of the operator matrix $\mathbf{A}_h(\boldsymbol{\mu})$, the effectivity upper bound is in fact the condition number of the high-fidelity problem, that is it measures the sensitivity of the high-fidelity problem with respect to small perturbations. Therefore, the effectivity upper bound is independent of N and hence, stable with respect to N-refinement. However, independently of the reduced approximation, we can expect large effectivities when the underlying high-fidelity problem is ill-conditioned.

2.2.2.5 Empirical interpolation method

As seen in Section 2.2.2.2, the computational efficiency of the RB method relies on the affine decomposition (2.69). While in the linear case the assumption of affine parametric dependence proved to be sufficient to deliver computational efficiency, in the nonlinear case this turns out to be only a necessary condition. Indeed, if the residual $r(\cdot;\boldsymbol{\mu})$ in (2.87) is not affine in the parameter space, the online complexity will no longer be independent of N_h and the dimensionality reduction will not necessarily imply CPU reduction and as a result, will not admit an efficient online-offline decomposition. Hence, a further level of reduction called *hyper-reduction* will be introduced, suitably employing techniques such as EIM, which recovers online N_h independence even in the presence of non-affine parameter dependecy. This method was first introduced in [21] and in the context of ROM in [61]. Some applications of the EIM method are discussed in [97] and an a posteriori error analysis is presented in [51, 61].

Consider a parameter-dependent family of functions $\mathcal{H} = \{h(\cdot;\boldsymbol{\mu}); \boldsymbol{\mu} \in \mathcal{P}\}$ which has to belong to $C^0(\bar{\Omega})$ since interpolation procedures requires point-wise evaluations of the functions $h(\cdot;\boldsymbol{\mu})$. The approximation is obtained through an interpolation operator $\mathcal{I}_M^{\mathbf{x}}$ that interpolates the function $h(\cdot;\boldsymbol{\mu})$ at some particular interpolation points $X_M = \{\mathbf{x}^1, \dots, \mathbf{x}^M\} \subset \bar{\Omega}$ frequently called *magic points* as a linear combination of some carefully chosen basis functions $\{\rho_1, \dots, \rho_M\}$. The superscript $\mathbf{x}$ of the interpolation operator represents the fact that the interpolation is performed with respect to $\mathbf{x}$. Then, the interpolant $\mathcal{I}_M^{x} h(\cdot;\boldsymbol{\mu})$ of $h(\cdot;\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathcal{P}$ admits the separable expansion

$$\mathcal{I}_M^{\mathbf{x}} h(\mathbf{x};\boldsymbol{\mu}) = \sum_{j=1}^{M} c_j(\boldsymbol{\mu}) \rho_j(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{2.92}$$

and satisfies the M interpolation constrains

$$\mathcal{I}_M^{\mathbf{x}} h(\mathbf{x}^i;\boldsymbol{\mu}) = h(\mathbf{x}^i;\boldsymbol{\mu}), \quad i = 1, \dots, M. \tag{2.93}$$

This yields to the following system that has to be solved

$$\sum_{j=1}^{M} c_j(\boldsymbol{\mu}) \rho_j(\mathbf{x}^i) = h(\mathbf{x}^i;\boldsymbol{\mu}), \quad i = 1, \dots, M \quad \text{or in matrix form} \tag{2.94}$$

$$\mathbf{T}_M \mathbf{c}(\boldsymbol{\mu}) = \mathbf{h}_M(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathcal{P}, \tag{2.95}$$

where $(\mathbf{T}_M)_{ij} = \rho_j(\mathbf{x}^i)$, $(\mathbf{c}(\boldsymbol{\mu}))_j = c_j(\boldsymbol{\mu})$ and $(\mathbf{h}_M(\boldsymbol{\mu}))_i = h(\mathbf{x}^i;\boldsymbol{\mu})$ for $i, j = 1, \dots, M$.

We start to construct the basis functions and the interpolation points based on a greedy algorithm in which we add the particular function h that is least well approximated by the current interpolation operator. Let's start by choosing the first sample point as

$$\boldsymbol{\mu}^1_{EIM} = \arg\max_{\boldsymbol{\mu}\in\mathcal{P}} ||h(\cdot;\boldsymbol{\mu})||_{L^\infty(\Omega)},$$

define the sample parameter points $S_1 = \{\boldsymbol{\mu}^1_{EIM}\}$ and then generate the first function as

$$\xi_1(\mathbf{x}) = h(\mathbf{x};\boldsymbol{\mu}^1_{EIM}).$$

Concerning the interpolation nodes, we first set

$$\mathbf{x}^1 = \arg\max_{\mathbf{x}\in\bar{\Omega}} |\xi_1(\mathbf{x})|, \quad X_1 = \{\mathbf{x}^1\};$$

then we define the first basis function as

$$\rho_1(\mathbf{x}) = \xi_1(\mathbf{x})/\xi_1(\mathbf{x}^1)$$

and set $V_1 = \mathrm{span}\{\rho_1\}$. Finally, we set the interpolation matrix

$$(\mathbf{T}_M)_{11} = \rho_1(\mathbf{x}^1) = 1.$$

At the m-th step, $m = 1,\ldots,M-1$, given the (nested) set of interpolation points $X_M = \{\mathbf{x}^1,\ldots,\mathbf{x}^M\}$ and the set $\{\rho_1,\ldots,\rho_m\}$ of basis functions, we select as the $(m+1)$-th generating function the snapshot which is the worst approximated by the current interpolant, i.e we select the snapshot which maximizes the error between h and its current interpolant $\mathcal{I}^{\mathbf{x}}_m h$:

$$\boldsymbol{\mu}^{m+1}_{EIM} = \arg\max_{\boldsymbol{\mu}\in\mathcal{P}} ||h(\cdot;\boldsymbol{\mu}) - \mathcal{I}^{\mathbf{x}}_m h(\cdot;\boldsymbol{\mu})||_{L^\infty(\Omega)},$$

$$\xi_{m+1}(\mathbf{x}) = h(\mathbf{x};\boldsymbol{\mu}^{m+1}_{EIM})$$

and we set $S_{m+1} = S_m \cup \{\boldsymbol{\mu}^{m+1}_{EIM}\}$.

To choose the $(m+1)$-th interpolation point, we first evaluate the residual

$$r_{m+1}(\mathbf{x}) = \xi_{m+1}(\mathbf{x}) - \mathcal{I}^{\mathbf{x}}_m \xi_{m+1}(\mathbf{x})$$

by solving the linear system

$$\sum_{j=1}^{m} c_j(\boldsymbol{\mu})\rho_j(\mathbf{x}^i) = \xi_{m+1}(\mathbf{x}^i), \quad i = 1,\ldots,m$$

to characterize the interpolant $\mathcal{I}^{\mathbf{x}}_m \xi_{m+1}$; then, we set

$$\mathbf{x}^{M+1} = \arg\max_{\mathbf{x}\in\bar{\Omega}} |r_{m+1}(\mathbf{x})|$$

that is, the point of Ω where ξ_{m+1} is the worst approximated. Finally, the new basis function is defined as

$$\rho_{m+1}(\mathbf{x}) = \frac{r_{m+1}(\mathbf{x})}{r_{m+1}(\mathbf{x}^{m+1})}$$

and we set $V_{m+1} = \mathrm{span}\{\rho_i, i = 1,\ldots,m+1\}$. The EIM algorithm is performed until a given tolerance is reached or until a given number of terms is computed and it yields to a sequence of hierarchical spaces $V_1 \subset V_2 \subset \ldots V_M$, such that the interpolation is exact for any $v \in V_M$ i.e $\mathcal{I}^{\mathbf{x}}_M v = v, \forall v \in V_M$ provided that $\dim(V_M) = M$ and the matrix $\mathbf{T}_M \in \mathbb{R}^{M\times M}$ is invertible.

Chapter 3: Model order reduction using L^1-norm minimization

3.1 Introduction

Model reduction is becoming an essential tool to enable applications requiring either real-time predictions or the evaluation of a large number of partial differential equations (PDE) based computational models. The first category encompasses optimal control [73, 95] and model predictive control [13, 70]. Routine analysis and parametrized studies [12], design optimization [15, 87] and the quantification of uncertainty [32] are applications pertaining to the second category, to name just a few. In all of these applications, the large dimensionality associated with the discretized partial differential equations prevents their solution in real-time. Model reduction reduces that cost by restricting the solution to a subspace of the solution space. This subspace is usually described by a small number of reduced basis vectors. In turn, a projection step reduces the dimensionality of the system of discrete equations considered, enabling their fast solution.

Model order reduction of elliptic and parabolic PDEs has been the subject of numerous studies [81, 136] and its theory is well understood [22, 23, 62, 69, 82, 113, 122], even though some elliptic problems such as problems in solid mechanics involving severe heterogenities, contact, or simply multi-point constraints are not very easy to deal with. Related to convection dominated or hyperbolic problems, the RB technique itself has not so often been applied, because moving waves and discontinuities such as shocks require a large number of basis vectors to accurately approximate these features [44]. This characterizes these problems as ones with large Kolmogorov N-widths [27]. Nevertheless, in this context of parameterized nonlinear evolution equations, we mention the following papers, each of them proposing different methods in order to deal with nonlinearities, using: approximations of generalized Lax pairs [53], the solution of Monge-Kantorovich mass transfer problem [72], the method of freezing [106] or using the domain partitioning method, followed by an interpolation step [128].

All this prior work is mostly based on the studies and the theory of model order reduction for elliptic and parabolic equations and using an L^2-minimization norm. In this chapter, we are using the L^1-norm minimization for determining the generalized coordinates, as an alternative to the L^2-norm. This norm is very closely linked to the concept of weak solution of hyperbolic conservation laws, obtaining a non-oscillatory behavior of the numerical solution.

In order to reduce the Kolmogorov N-width, approaches based on local bases considering local subspaces can be found in [14, 16, 48, 99]. The locality can be characterized in parameters [14], time [48] or state-space [16]. In the present work, an approach based on dictionaries is considered [30, 78]. More specifically, solutions corresponding to various time and parameter instances are collected and stored in such a dictionary using a greedy sampling method [32, 62, 136]. Each solution will then be considered as a reduced basis vector. In turn, localization in time and space can be easily enforced by only considering basis vectors corresponding to restricted sub-domains of the time and parameters spaces. In addition to the reduction in number of basis vectors, this thesis will demonstrate that a key advantage of a dictionary approach is a better approximation of states having sharp gradients and discontinuities. In particular, it will be demonstrated that avoiding basis truncation such as the one occurring in Proper Orthogonal Decomposition (POD) [127] or Non-Negative Matrix Factorization [18] avoid Gibbs phenomenon.

In addition to the choice of reduced basis, a key ingredient in projection-based model reduction is the definition of the reduced system of equations. For symmetric systems such as those arising in elliptic and parabolic PDEs, Galerkin projection is the method of choice. For non-symmetric systems, it has been shown that minimizing the L^2-norm of the residual is preferable for stability, unicity and optimality considerations [36, 37]. Nevertheless, in this thesis, we will present a minimum residual approach for determining the generalized coordinates by using the L^1-norm (adding a regularization and a perturbation term to it) as an alternative to the L^2-norm and we will show it's robustness in the context of hyperbolic problems. More specifically, the present work demonstrates that combining a dictionary and L^1-minimization promotes sparsity in the choice of basis functions, participating in the reduced-order solution and resulting in accurate and physical reduced-order solutions.

In here, we develop robust error estimators, select the dictionary elements using a greedy sampling algorithm, we illustrate the advantage of using L^1-norm minimization on a one and two dimensional example (2D) and we make use of hyper-reduction ideas, discussing also the cost of the method. Moreover, in the current work, we are giving implementation details of the method, we give solutions to the potential difficulties that might arise when using L^1-norm minimization and last but not least, in the numerical applications part, we are comparing different L^1-minimization approaches with the L^2-norm minimization, we study the convergence and the sparsity of the method and we illustrate for different targets, the quality of the reconstructed solution based on the number of elements in the dictionary.

This chapter is organized as follows: we first discuss the problem of interest. In the following section, an approximation of the solution of nonlinear problems by reduced order models is presented. In Section 3.4, we explain the role of L^1-norm minimization in this problem, then we present in detail the algorithm we have developed and provide an error estimate. In Section 3.6 we describe the greedy sampling algorithm, used to select the elements in the reduced basis. In Section 3.7 we describe the potential difficulties that can arise using L^1-norm minimization, we give solutions how to fix them and in the end we present efficient algorithms for the computation of the L^1-norm minimization, both in the cases of linear and nonlinear residuals. In Section 3.8, we are discussing the computational cost of the method. The last section provides several numerical examples that illustrate the behavior of our methods, on linear and nonlinear problems, both in one (1D) and two dimensional case.

3.2 Problem of interest

In this work, high-dimensional models (HDM) arising from the space discretization of hyperbolic PDEs are considered. PDEs of the following type are considered

$$\begin{cases} \dfrac{\partial \mathbf{W}}{\partial t} + \mathbf{L}(\mathbf{W};\boldsymbol{\mu}) & = \mathbf{f}(t,\boldsymbol{\mu}),\ \mathbf{x}\in\Omega,\ t\in[0,T], \\ \mathbf{B}(\mathbf{W};\boldsymbol{\mu}) & = \mathbf{g}(t,\boldsymbol{\mu}),\ \mathbf{x}\in\partial\Omega,\ t\in[0,T], \\ \mathbf{W}(\mathbf{x},t=0,\boldsymbol{\mu}) & = \mathbf{W}_0(\mathbf{x},\boldsymbol{\mu}),\ \mathbf{x}\in\Omega, \end{cases} \tag{3.1}$$

where $\mathbf{W} = \mathbf{W}(\mathbf{x}, t) \in \mathbb{R}^m$ is the state variable, $t \in [0, T]$ is the time variable, $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ is the space variable ($1 \leq d \leq 3$) and $\partial\Omega$ is the boundary of the domain. $\mathbf{L}$ is a differential operator such as the divergence of a flux and $\mathbf{B}$ a boundary operator, $\mathbf{f}$ and $\mathbf{g}$ are volume and surface forces, respectively and $\boldsymbol{\mu} = (\mu^1, \cdots, \mu^p) \in \mathcal{P} \subset \mathbb{R}^p$ is a vector of p parameters defining the system of interest.

Discretizing the PDE (3.1) using finite differences approximation or finite volume formulation leads to a system of large dimension $N = m \times N_{space}$ of ordinary differential equations (ODEs) of the following form

$$\begin{cases} \dfrac{d\mathbf{w}}{dt} + \mathbf{f}(\mathbf{w}, t; \boldsymbol{\mu}) & = \mathbf{g}(t, \boldsymbol{\mu}), \ t \in [0, T] \\ \mathbf{w}(t = 0, \boldsymbol{\mu}) & = \mathbf{w}_0(\boldsymbol{\mu}), \end{cases} \tag{3.2}$$

where $\mathbf{w} = \mathbf{w}(t, \boldsymbol{\mu}) \in \mathbb{R}^N$ is the HDM state, t denotes the time and $\mathbf{f}(\cdot, \cdot)$, $\mathbf{g}(\cdot)$ are nonlinear functions of their arguments. This problem is supplemented with boundary conditions which are specified in the last section of this chapter.

In the remainder of this chapter, the time and parameter variables are grouped together, unless explicitly stated, as a variable $\boldsymbol{\tau} = [t; \boldsymbol{\mu}]$. Hence, the HDM state is parametrized as

$$\mathbf{w}(\boldsymbol{\tau}) = \mathbf{w}(t, \boldsymbol{\mu}). \tag{3.3}$$

In practice, the ODE (3.2) is discretized in time using a time discretization $t_0 = 0 < t_1 < \cdots < t_{N_t} = T$. Explicit and implicit time-discretization techniques are used in the present chapter, resulting in a sequence of nonlinear systems of equations of large dimension N

$$\mathbf{r}^n(\mathbf{w}) = \mathbf{0}, \ n = 1, \cdots, N_t, \tag{3.4}$$

where $\mathbf{r}^n = [r_1^n, \cdots, r_N^n]^T$. We give several examples later in the text. Note that the residual $\mathbf{r}^n$ will depend on several time instances of the solution for unsteady problems, for example $\mathbf{w}^n$ and $\mathbf{w}^{n-1}$ in the simplest case. Steady problems can also be written in the form $\mathbf{r}(\mathbf{w}) = \mathbf{0}$.

The goal of model reduction is to approximate the high-dimensional system (3.4) using a much smaller number of variables while retaining accuracy of the solution. For that purpose, projection-based model reduction techniques approximate the state $\mathbf{w}(\boldsymbol{\tau})$ in a subspace of $\mathbb{R}^N$ using a reduced-order basis (ROB)/dictionary $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathbb{R}^{N \times k}$, $k << N$. The state is then approximated as

$$\mathbf{w}(\boldsymbol{\tau}) \approx \mathbf{V}\mathbf{q}(\boldsymbol{\tau}) = \sum_{i=1}^{k} \mathbf{v}_i q_i(\boldsymbol{\tau}) \tag{3.5}$$

where $\mathbf{q}(\boldsymbol{\tau}) = [q_1(\boldsymbol{\tau}), \cdots, q_k(\boldsymbol{\tau})]^T \in \mathbb{R}^k$ denotes the vector of k reduced coordinates. Substituting the subspace approximation (3.5) into (3.4) usually results in a non-zero residual of dimension N

$$\mathbf{r}^n(\mathbf{V}\mathbf{q}) \approx \mathbf{0}. \tag{3.6}$$

which accounts for the fact that $\mathbf{V}\mathbf{q}(\boldsymbol{\tau})$ is not in general an exact solution of the dynamical equation. Two common approaches result in the definition of a reduced system of equations:

- Galerkin projection [81, 82] enforces the orthogonality of the residual to the ROB $\mathbf{V}$ as

$$\mathbf{V}^T \mathbf{r}^n(\mathbf{V}\mathbf{q}) = \mathbf{0},\ n = 1, \cdots, N_t. \tag{3.7}$$

  This defines a set of k nonlinear equations in terms of k unknowns which can be solved by Newton-Raphson's method.

- Residual minimization approaches [16,32,36,37,87] minimize the residual in the L^2-norm sense

$$\min_{\mathbf{q} \in \mathbb{R}^k} \|\mathbf{r}^n(\mathbf{V}\mathbf{q})\|_2^2 = \sum_{i=1}^{N} \left(r_i^n(\mathbf{V}\mathbf{q})\right)^2,\ n = 1, \cdots, N_t. \tag{3.8}$$

  In practice, this nonlinear least-squares problem can be solved using Gauss-Newton or Levenberg-Marquardt iterations [105]. In Section 3.7, alternative residual minimization approaches based on L^1-norm minimization which are more appropriate for the reduction of hyperbolic problems will be proposed.

3.3 Dictionary approach

Projection-based model reduction techniques [18, 122, 127] based on the pre-computed snapshots of the HDM for specific values of the vector $\boldsymbol{\tau} = [t; \boldsymbol{\mu}]$. These snapshots are gathered in a snapshot matrix

$$\mathbf{S} = [\mathbf{w}(\boldsymbol{\tau}_1), \cdots, \mathbf{w}(\boldsymbol{\tau}_{N_s})]. \tag{3.9}$$

Five approaches for compressing the snapshot matrix are described as follows:

- Proper Orthogonal Decomposition [127] computes an optimal reduced-order basis of dimension k that minimizes the projection error of the snapshots onto the basis.
- Balanced POD [140], applicable to linear systems only, also takes into account snapshots of the dual system to construct the reduced basis for the primal and dual systems.
- Non-negative matrix factorization [86] was recently applied to construct a non-negative reduced-order basis based on snapshots with positive entries in the context of contact problems [18]. The reduced basis minimizes the positive reconstruction of the snapshots.
- POD-Greedy algorithm is a combination of the Greedy algorithm with a temporal compression step. The main ingredient for time-sequence compression is the use of POD [66].

All four approaches perform a compression of the information contained in the snapshot matrix $\mathbf{S}$. More specifically, the N_s vectors contained in $\mathbf{S}$ are compressed, leading to a reduced-order basis of dimension $k \leq N_s$.

In essence, RB methods are based on a two step strategy: the first step (offline stage) allows selecting particular instances of the parameters, for which a very accurate approximation of the solution is computed: those solutions constitute the reduced basis. In a second step (the online stage), the generic solutions (for other instances of the parameter) are approximated by a linear combination of the reduced basis functions.

In this contribution, we investigate problems with very high sensitivity with respect to the parameter $\boldsymbol{\mu}$ that yield unfeasibly large reduced bases. For these kinds of problems, we introduce a new method for model order reduction that uses a dictionary of basis vector candidates to build a small basis during the offline phase. Our method holds some similarity with the locally adaptive Greedy method introduced in [99] and with the online greedy reduced basis construction using dictionaries introduced in [78]. The only difference is that in our method, the dictionary is constructed in an offline phase, based on an offline greedy algorithm. This means, that the dictionary is constructed offline, in an iterative way, by finding the parameter $\boldsymbol{\mu}$ worst approximated in the current dictionary and enrich the current dictionary with $\mathbf{w}(t, \boldsymbol{\mu})$. In the online phase, this dictionary will combine good approximation quality and will constitute a small online basis.

Hence, an approach based on a dictionary of solutions is used, as it does not incur any loss of information by compression, for which a lot of modes are needed for the reconstruction, especially for high-dimensional problems (in the simple scalar advection problem (1.5), it was needed 150 modes to reduce with three orders of magnitude the energy). As such, the vectors $\{\mathbf{v}_i\}_{i=1}^k$ in the reduced basis are the solutions of the HDM:

$$\mathbf{v}_i = \mathbf{w}(\boldsymbol{\tau}_i) = \mathbf{w}(t, \boldsymbol{\mu}_i), \;\; i = 1, \ldots, k \tag{3.10}$$

and these vectors also constitute our dictionary. We denote with $\mathcal{D} = \{\boldsymbol{\mu}_i\}_{i=1}^k$ the set of parameters chosen by the Greedy Algorithm 1 from a big training set $\mathcal{C} = \{\boldsymbol{\mu}_i\}_{i=1}^{N_c}$, $k < N_c$ at each greedy iteration. So, the number of samplings corresponds to the number of elements in the reduced basis/dictionary. In this case, the error can be controlled using the error estimate given in Section 3.5.2. In the end, when a RB approximation is to be computed for a certain given parameter $\boldsymbol{\mu} \in \mathcal{P}$ in the online stage, we only use the k precomputed basis functions i.e the solution of the HDM will then be approximated as

$$\mathbf{w}(\boldsymbol{\tau}) \approx \sum_{i=1}^{k} \mathbf{w}(\boldsymbol{\tau}_i) q_i(\boldsymbol{\tau}). \tag{3.11}$$

In the present case, since the HDM is of very large dimension, over-complete dictionaries, as used in compressed sensing [35, 49] and for which $k \geq N$ will not be considered.

3.4 L^1-norm residual minimization

In the present section, model reduction based on L^1-norm residual minimization is introduced to reduce the dimensionality of hyperbolic equations as an alternative to Galerkin projection and L^2-norm minimization. Motivations for the use of the L^1-norm are provided in this section. Model reduction based on L^1-norm minimization will be introduced in Section 3.7 together with practical numerical procedure for their computation in Section 3.7.2.

Minimizing the L^1-norm of the residual is known to lead to regressions that are much more robust to outliers [28]. In the context of hyperbolic systems, Lavery's work [83] was probably the first one which used L^1-minimization to solve hyperbolic problems. It was followed by Guermond et al. on Hamilton Jacobi equations and transport problems [63, 64] and it has been shown, at least experimentally, that the numerical solution can retain an excellent non-oscillatory behavior by minimizing the L^1-norm of the PDE residual. In [64], the schemes are designed by minimizing quantities that mimic the total variation of a functional, as in here. In the following sections, the idea is exploited in the case of model reduction. For completeness, the motivation for L^1-norm minimization is justified as follows for the problem

$$\frac{\partial \mathbf{W}}{\partial t} + \text{ div } \mathbf{F}(\mathbf{W}) = 0 \tag{3.12}$$

defined on $\Omega \subset \mathbb{R}^d \times \mathbb{R}^+$. The solution $\mathbf{W} = \mathbf{W}(\mathbf{x}, t)$ belongs here to $\mathbb{R}^m$, so that $\mathbf{F} = (F_1, \ldots, F_m)^T$. The weak form of the equation is: for any $\boldsymbol{\varphi} \in \left[C_0^1(\mathring{\Omega})\right]^m$ with compact support in the interior $\mathring{\Omega}$ of Ω:[1]

$$\int_\Omega \boldsymbol{\varphi}(\mathbf{x}, t)\left(\frac{\partial \mathbf{W}}{\partial t} + \text{ div } \mathbf{F}(\mathbf{W})\right) dt d\mathbf{x} = 0. \tag{3.13}$$

Integrating by parts yields

$$\int_\Omega \frac{\partial \boldsymbol{\varphi}}{\partial t} \mathbf{W} dt d\mathbf{x} + \int_\Omega \nabla \boldsymbol{\varphi} \cdot \mathbf{F}(\mathbf{W}) dt d\mathbf{x} = 0. \tag{3.14}$$

Restricting to the set of test functions $\mathcal{T} = \left\{\boldsymbol{\varphi} \in \left[C_0^1(\mathring{\Omega})\right]^m, ||\boldsymbol{\varphi}||_\infty \leq 1\right\}$, $\mathbf{W}$ is a solution if:

$$\sup_{\boldsymbol{\varphi} \in \mathcal{T}} \left(\int_\Omega \frac{\partial \boldsymbol{\varphi}}{\partial t} \mathbf{W} dt d\mathbf{x} + \int_\Omega \nabla \boldsymbol{\varphi} \cdot \mathbf{F}(\mathbf{W}) dt d\mathbf{x} \right) = 0. \tag{3.15}$$

Remember that for any function $\mathbf{g} \in L^1(\mathbb{R}^d)$, the total variation is defined as

$$TV(\mathbf{g}) = \sup_{\boldsymbol{\varphi} \in C_0^1(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d), ||\boldsymbol{\varphi}||_\infty \leq 1} \left\{ \int_{\mathbb{R}^d} \nabla \boldsymbol{\varphi}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) d\mathbf{x} \right\}, \tag{3.16}$$

and if, in addition, $\mathbf{g} \in C^1(\mathbb{R}^d)$, then $TV(\mathbf{g}) = \int_{\mathbb{R}^d} ||\nabla \mathbf{g}|| dx = ||\nabla \mathbf{g}||_{L^1(\mathbb{R}^d)}$. This shows that, defining the space-time flux $\mathcal{F} = (\mathbf{W}, \mathbf{F})$, $\mathbf{W}$ is a weak solution if and only if the space-time total variation of $\mathcal{F}$ vanishes, that is

$$TV\big(\mathcal{F}(\mathbf{W})\big) = 0. \tag{3.17}$$

In other words, one can look for $\mathbf{W}$ as a function of $L^1 \cap L^\infty$ such that $\mathbf{W}$ minimizes $TV\big(\mathcal{F}(\mathbf{V})\big)$ over $\mathbf{V} \in L^1 \cap L^\infty$, i.e.

$$\mathbf{W} = \operatorname{argmin}\{TV\big(\mathcal{F}(\mathbf{V})\big), \mathbf{V} \in L^1 \cap L^\infty\}. \tag{3.18}$$

This does not guaranty uniqueness (and thus there is some abuse of language in this setting), since the entropy conditions are not encoded into this formulation. From (3.17), this non uniqueness can also be interpreted in term on the non strict convexity of the L^1 unit ball. However, (3.18) indicates that a natural setting is to minimize the L^1-norm of the space-time divergence of the space-time flux $\mathcal{F}$, but something must be added in order to generate unique solutions, as the entropy criteria is a way to guaranty uniqueness of the solution of (3.12).

[1] so that in particular $\forall (x, 0) \in \mathbb{R}^d \times \mathbb{R}^+$, $\varphi(x, 0) = 0$.

How does it translates to the discrete setting? For simplicity, we only mention the case of explicit schemes. We discuss later the solution procedure for the case of implicit schemes. The following classical result is mentioned. Consider $\{x_j\}_{j\in\mathbb{Z}}$ a strictly increasing sequence in $\mathbb{R}$ and $x_{j+1/2} = \frac{x_j+x_{j+1}}{2}$. Assuming that $\mathbb{R} = \cup_{j\in\mathbb{Z}}[x_{j-1/2}, x_{j+1/2}[$ and considering g defined as: for any $j \in \mathbb{Z}$,

$$g(x) = g_j \text{ if } x \in [x_{j-1/2}, x_{j+1/2}[, \tag{3.19}$$

then

$$TV(g) = \sum_{j\in\mathbb{Z}} |g_{j+1} - g_j|. \tag{3.20}$$

Consider now an approximation procedure that enables, from $\mathbf{w}^n \approx \mathbf{W}(\cdot, t_n)$, to compute $\mathbf{w}^{n+1} \approx \mathbf{W}(\cdot, t_{n+1})$. For instance, assume that we have a finite volume method, $d = 1$ and for any grid point $j \in \{1, \dots, N\}$, we define the mesh $x_{j+1/2} = j\Delta x$, $t^n = n\Delta t$ and the control volumes $c_j = (x_{j-1/2}, x_{j+1/2})$. Considering $\boldsymbol{\varphi} = \mathbf{1}_{[x_{j-1/2}, x_{j+1/2}]\times[t^n, t^{n+1}]}$ with $[x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1}] \subset \Omega$ in (3.13), we obtain the following:

$$\begin{aligned}
0 &= \int_{x_{j-1/2}}^{x_{j+1/2}} \int_{t^n}^{t^{n+1}} \left(\frac{\partial \mathbf{W}}{\partial t} + \text{ div } \mathbf{F}(\mathbf{W})\right) dt dx \\
&= \Delta x \int_{x_{j-1/2}}^{x_{j+1/2}} \frac{1}{\Delta x} \mathbf{W}(x, t^{n+1}) dx - \Delta x \int_{x_{j-1/2}}^{x_{j+1/2}} \frac{1}{\Delta x} \mathbf{W}(x, t^n) dx \\
&\quad + \Delta t \int_{t^n}^{t^{n+1}} \frac{1}{\Delta t} \mathbf{F}(\mathbf{W}(x_{j+1/2}, t)) dt - \Delta t \int_{t^n}^{t^{n+1}} \frac{1}{\Delta t} \mathbf{F}(\mathbf{W}(x_{j-1/2}, t)) dt.
\end{aligned}$$

Using the approximations, $\mathbf{w}_j^n \approx \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{W}(x, t^n) dx$ and $\mathbf{f}_{j+1/2}(\mathbf{w}^n) \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{W}(x_{j+1/2}, t)) dt$, we obtain:

$$\Delta x(\mathbf{w}_j^{n+1} - \mathbf{w}_j^n) + \Delta t\big(\mathbf{f}_{j+1/2}(\mathbf{w}^n) - \mathbf{f}_{j-1/2}(\mathbf{w}^n)\big) = 0. \tag{3.21}$$

In this case, the residual can be written as:

$$\big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_j = \mathbf{w}_j^{n+1} - \mathbf{w}_j^n + \frac{\Delta t}{\Delta x}\big(\mathbf{f}_{j+1/2}(\mathbf{w}^n) - \mathbf{f}_{j-1/2}(\mathbf{w}^n)\big) \tag{3.22}$$

and we can evaluate the value of $\mathbf{w}^{n+1}$ as the minimization solution of the total variation:

$$TV(\boldsymbol{r}) = \sum_{j=1}^{N} \Big| \big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_j \Big|. \tag{3.23}$$

In (3.22), $\mathbf{f}_{j+1/2}$ is any consistent numerical flux at the cell interface $x_{j+1/2}$; see [131] for the classical examples.

Substituting (3.5) in (3.22), for all time steps, find the coefficients q_i^n that minimizes the following residual:

$$\boldsymbol{r}_j^n(\boldsymbol{V}\boldsymbol{q}) = \sum_{i=1}^{k} q_i^{n+1} \mathbf{w}_j^{n+1}(\boldsymbol{\mu}_i) - \sum_{i=1}^{k} q_i^n \mathbf{w}_j^n(\boldsymbol{\mu}_i) + \frac{\Delta t}{\Delta x}\left(\mathbf{f}_{i+1/2}\big(\sum_{i=1}^{k} q_i^n \mathbf{w}_j^n(\boldsymbol{\mu}_i)\big) - \mathbf{f}_{i-1/2}\big(\sum_{i=1}^{k} q_i^n \mathbf{w}_j^n(\boldsymbol{\mu}_i)\big)\right). \tag{3.24}$$

For the case of implicit schemes, the solution is obtained similarly as for the explicit case, obtaining the following residual that has to be minimized:

$$\boldsymbol{r}_j^n(\boldsymbol{V}\boldsymbol{q}) = \sum_{i=1}^{k} q_i^{n+1}\mathbf{w}_j^{n+1}(\boldsymbol{\mu}_i) - \sum_{i=1}^{k} q_i^{n}\mathbf{w}_j^{n}(\boldsymbol{\mu}_i) + \frac{\Delta t}{\Delta x}\Bigg(\mathbf{f}_{i+1/2}\big(\sum_{i=1}^{k} q_i^{n+1}\mathbf{w}_j^{n+1}(\boldsymbol{\mu}_i)\big) - \mathbf{f}_{i-1/2}\big(\sum_{i=1}^{k} q_i^{n+1}\mathbf{w}_j^{n+1}(\boldsymbol{\mu}_i)\big)\Bigg). \tag{3.25}$$

Remark 3.4.1. *The L^1-norm is convex but not strictly convex, and hence the minimization problem* (3.23) *may not have a unique solution. For this reason, in practice, we perturb the functional* (3.23) *to make it strictly convex. Let us denote it by J, and thus we will look for solutions that minimize*

$$J(\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})). \tag{3.26}$$

Examples are:

1. *for $\nu > 0$,*

$$J(\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})) = \sum_{j=1}^{N} \Big| \big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_j \Big| + \nu \sum_{j=1}^{N} (\mathbf{w}_j^{n+1})^2, \tag{3.27a}$$

2. *More generally, if U is a convex entropy,*

$$J(\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})) = \sum_{j=1}^{N} \Big| \big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_j \Big| + \nu \sum_{j=1}^{N} U(\mathbf{w}_j^{n+1}). \tag{3.27b}$$

   The functional (3.27b) *can be used for systems and the choice of $\nu > 0$ will be discussed in Section 3.7.1.*

Another example of strictly convex functionals, probably more linked to the characterization of entropy solution, is: for $\nu > 0$,

$$J(\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})) = \sum_{j=1}^{N} \Big| \big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_j \Big| + \nu \sum_{j=1}^{N} (\mathbf{w}_j^{n+1} - \mathbf{w}_j^{n})^2.$$

Remark 3.4.2. *Other potential difficulties using the minimization of a L^1-norm will be presented in Section 3.7.*

3.5 Error estimation

In this section, we provide an error estimate (in the scalar case) between the solution obtained by projecting onto the span of the dictionary or onto the convex hull of the dictionary and the solution of the original scheme. These error estimates are another way to justify the method and are provided in a simple setting: we consider a monotone scheme. In this section, we first precise the scheme settings, then we give a natural condition on the dictionary for obtaining these error estimates and in the end we state them and prove them.

3.5.1 Scheme setting

Consider the scalar conservation law equations with the initial condition:

$$\begin{aligned} \frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} &= 0, \quad x \in \mathbb{R}, t > 0 \\ w(x,0) &= w_0(x), \quad x \in \mathbb{R}. \end{aligned} \tag{3.28}$$

After discretizing, we assume that the scheme writes, for $w := (w_j)_{j\in\mathbb{Z}}$,

$$w_j^{n+1} = S(w_j^n, \lambda) \tag{3.29}$$

with $\lambda = \Delta t/\Delta x$ and the initial condition

$$w_j^0 = \text{given}. \tag{3.30}$$

We assume that the operator S is monotone for $\lambda \in [0, b[$, $b > 0$, i.e. if for any sequence w and v bounded for the L^1 or L^∞-norms with $j \in \mathbb{Z}$, $w_j \le v_j$, then $S(w_j, \lambda) \le S(v_j, \lambda)$. Let L^1 and L^∞-norms be generically denoted by $||\cdot||$.

An example is given by the explicit scheme

$$S(w_j^n, \lambda) = w_j^n - \lambda\big(\hat{f}(w_{j+1}^n, w_j^n) - \hat{f}(w_j^n, w_{j-1}^n)\big) \tag{3.31}$$

where we assume that the numerical flux $\hat{f}(a, b)$ is monotone, i.e. increasing with respect to the first variable and decreasing with respect to the second one and the operator S is monotone under a CFL like condition.

Another example is given by the implicit scheme, where w_j^n is defined as the solution of

$$S(w_j^{n+1}, \lambda) = w_j^{n+1} + \lambda\big(\hat{f}(w_{j+1}^{n+1}, w_j^{n+1}) - \hat{f}(w_j^{n+1}, w_{j-1}^{n+1})\big) \tag{3.32}$$

which is unconditionally monotone.

Thanks to Crandall-Tartar lemma (for example, see [56]), we know that for any w and v,

$$||S(w, \lambda) - S(v, \lambda)||_{L_1} \le ||w - v||_{L_1}.$$

The same is true in the L^∞ and L^2-norms.

3.5.2 Error estimate

We collect and store into the dictionary $\mathbf{V}$ the solutions $\{w^n(\boldsymbol{\mu}_i)\}_i$ of the problem (3.28) which correspond to various time and parameter instances and where also the initial conditions are depending on these parameters $\{\boldsymbol{\mu}_i\}_{i=1,\dots,k} \in \mathcal{D}$. Since the minimization procedure admits a unique solution, this enables to define a projection operator p^n for any time t^n, by solving the following minimization problem: knowing $w_\star^n \in \operatorname{span}_{\boldsymbol{\mu}_i\in\mathcal{D}} (\{w^n(\boldsymbol{\mu}_i)\})$, find $w_\star^{n+1} \in \operatorname{span}_{\boldsymbol{\mu}_i\in\mathcal{D}} (\{w^{n+1}(\boldsymbol{\mu}_i)\})$ such that for any target $\boldsymbol{\mu} \in \mathcal{P}$,

$$w_\star^{n+1}(\boldsymbol{\mu}) := \underset{v_\star \in \underset{\boldsymbol{\mu}_i \in \mathcal{D}}{\text{span}} (\{w^{n+1}(\boldsymbol{\mu}_i)\})}{\text{argmin}} \big\{ J\big(v_\star - S(w_\star^n(\boldsymbol{\mu}), \lambda)\big)\big\} = p^n\big(S(w_\star^n(\boldsymbol{\mu}), \lambda)\big),$$

with J strictly convex. We consider in the following estimates that the scheme S is explicit. The same can be proven also for an implicit scheme.

Remembering that (3.29) applies for the elements of the dictionary,we have immediately the following estimate:

$$\begin{aligned}
J\bigg(w_\star^{n+1}(\boldsymbol{\mu}) - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) &= J\bigg(p^n(S(w_\star^n(\boldsymbol{\mu}), \lambda)) - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) \\
&= \min_{v_\star \in \underset{\boldsymbol{\mu}_i \in \mathcal{D}}{\text{span}} (\{w^{n+1}(\boldsymbol{\mu}_i)\})} J\bigg(v_\star - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) \\
&\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\bigg(w^{n+1}(\boldsymbol{\mu}_i) - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) \\
&= \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\bigg(S(w^n(\boldsymbol{\mu}_i), \lambda) - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) \\
&\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\bigg(w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})\bigg)
\end{aligned} \tag{3.33}$$

provided λ enables to fulfill the monotonicity property for all the elements of the dictionary. In (3.33), the passage from line 2 to 3 simply comes from the fact we are minimizing on a smaller subset of $\underset{\boldsymbol{\mu}_i \in \mathcal{D}}{\text{span}} (\{w^{n+1}(\boldsymbol{\mu}_i)\})$, namely the element dictionary. The last inequality in (3.33) relies on (3.29). It is possible due because of the monotonicity of the scheme for the functionals defined in Remark 3.4.1 since a monotone scheme is L^1 stable, L^2 stable, and TV stable [56].

Next, we consider the case where we project on the convex envelop of the dictionary, i.e

$$\min_{\mathbf{q} \in \mathbb{R}^k} \|\mathbf{r}^n(\mathbf{V}\mathbf{q})\|_1 \quad \textit{subject} \quad \textit{to} \quad \mathbf{1}^T q = 1, q \geqslant 0, \ n = 1, \cdots, N_t. \tag{3.34}$$

So the projection is a convex combination of the elements in the dictionary and for any target $\boldsymbol{\mu} \in \mathcal{P}$ we define:

$$w_\star^{n+1}(\boldsymbol{\mu}) = p^n(S(w_\star^n(\boldsymbol{\mu}), \lambda)) = \sum_{i=1}^{k} \alpha_i^{n+1} w^{n+1}(\boldsymbol{\mu}_i)$$

with $\alpha_i^n \geq 0$ and $\sum\limits_{i=1}^{k} \alpha_i^n = 1, \ \forall n.$

We obtain a sharper error estimate of type:

$$J\bigg(w_\star^{n+1}(\boldsymbol{\mu}) - S(w_\star^n(\boldsymbol{\mu}), \lambda)\bigg) \leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\bigg(w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})\bigg) \tag{3.35}$$

$$= \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\bigg(w^n(\boldsymbol{\mu}_i) - \sum_{j=1}^{k} \alpha_j^n w^n(\boldsymbol{\mu}_j)\bigg) \tag{3.36}$$

$$\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \left(\sum_j^k |\alpha_j^n| \right) \max_{\boldsymbol{\mu}_j \in \mathcal{D}} J\Big(w^n(\boldsymbol{\mu}_i) - w^n(\boldsymbol{\mu}_j) \Big) \tag{3.37}$$

$$= \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \max_{\boldsymbol{\mu}_j \in \mathcal{D}} J\Big(w^n(\boldsymbol{\mu}_i) - w^n(\boldsymbol{\mu}_j) \Big) \tag{3.38}$$

$$\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \max_{\boldsymbol{\mu}_j \in \mathcal{D}} J\Big(w^0(\boldsymbol{\mu}_i) - w^0(\boldsymbol{\mu}_j) \Big) =: \mathcal{J}(\mathcal{D}). \tag{3.39}$$

Again, the last inequality in (3.35) is possible due to the monotonicity of the scheme for the functionals defined in Remark 3.4.1 because a monotone scheme is L^1 stable, L^2 stable and TV stable [56]. We have shown the following result:

Proposition 3.5.1. *Consider J defined as in Remark 3.4.1. If $S(\cdot, \lambda)$ is a monotone scheme for $\lambda \in [0, b[$, $b > 0$ then:*

1. *At time t_{n+1} , the minimization is done onto the span of the dictionary $\{w^n(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i \in \mathcal{D}\}$, for any $\boldsymbol{\mu} \in \mathcal{P}$, the reduced solution $w_\star^{n+1}(\boldsymbol{\mu}) = p^n(S(w_\star^n(\boldsymbol{\mu}), \lambda))$ satisfies:*

$$J\Big(w_\star^{n+1}(\boldsymbol{\mu}) - S(w_\star^n(\boldsymbol{\mu}), \lambda) \Big) \leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\Big(w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu}) \Big).$$

2. *At time t_{n+1}, the minimization is done on the convex hull of the dictionary, for any $\boldsymbol{\mu} \in \mathcal{P}$, the reduced solution $w_\star^{n+1}(\boldsymbol{\mu}) = p^n(S(w_\star^n(\boldsymbol{\mu}), \lambda))$ satisfies:*

$$J\Big(w_\star^{n+1}(\boldsymbol{\mu}) - S(w_\star^n(\boldsymbol{\mu}), \lambda) \Big) \leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \max_{\boldsymbol{\mu}_j \in \mathcal{D}} J\Big(w^0(\boldsymbol{\mu}_i) - w^0(\boldsymbol{\mu}_j) \Big).$$

For J defined as in (3.27a) and using a projection over the span of the dictionary, we get the following estimate:

$$\begin{aligned}
J\Big(p^n(S(w_\star^n(\boldsymbol{\mu}), \lambda)) - S(w_\star^n(\boldsymbol{\mu}), \lambda) \Big) &= \min_{v_\star \in \underset{\boldsymbol{\mu}_i \in \mathcal{D}}{\text{span}} (\{w^{n+1}(\boldsymbol{\mu}_i)\})} J\Big(v_\star - S(w^n(\boldsymbol{\mu}), \lambda) \Big) \\
&\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\Big(w^{n+1}(\boldsymbol{\mu}_i) - S(w_\star^n(\boldsymbol{\mu}), \lambda) \Big) \\
&= \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\Big(S(w^n(\boldsymbol{\mu}_i), \lambda)) - S(w_\star^n(\boldsymbol{\mu}), \lambda) \Big) \\
&\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} J\Big(w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu}) \Big) \\
&= \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \left(||w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})||_1 + \nu ||w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})||_2^2 \right) \\
&\leq \min_{\boldsymbol{\mu}_i \in \mathcal{D}} \left[\left(1 + \nu ||w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})||_\infty \right) ||w^n(\boldsymbol{\mu}_i) - w_\star^n(\boldsymbol{\mu})||_1 \right]
\end{aligned}$$

where the last inequality holds due to the L^p-norms inequality: $||f||_q \leq ||f||_r^{r/q} ||f||_\infty^{1-r/q}$, with $q = 2$ and $r = 1$.

Using the same technique as for the proof of proposition 3.5.1, we obtain in the case of the convex hull projection the following estimate:

$$J\Big(p^n(S(w^n_\star(\boldsymbol{\mu}),\lambda)) - S(w^n_\star(\boldsymbol{\mu}),\lambda)\Big) \leq \min_{\boldsymbol{\mu}_i\in\mathcal{D}}\max_{\boldsymbol{\mu}_j\in\mathcal{D}}\Big[\big(1+\nu\|w^n(\boldsymbol{\mu}_i)-w^n(\boldsymbol{\mu}_j)\|_\infty\big) \\ \|w^n(\boldsymbol{\mu}_i)-w^n(\boldsymbol{\mu}_j)\|_1\Big].$$

Remark 3.5.2. *The error estimates described in Proposition 3.5.1, are only projection error estimates, as they do not account the modeling error but only the projection error.*

3.6 Training by greedy sampling

An essential step in the construction of a parametric ROM is the selection of the sampled snapshots in the time and parameter domains. Greedy approaches [32, 62, 136] proceed by iteratively selected the location in the parameter space where the error between the HDM and the ROM is the largest. Then, solve the full model for the chosen sampling and update the reduced model. These steps are repeated until the error is acceptable. As computing the error requires the expensive solution of the HDM, cheaper error indicators are used instead. In the present work, the cumulative L^1-norm of the residual vector corresponding to the ROM solution is used as error indicator:

$$\mathcal{E}(\boldsymbol{\mu}) = \frac{1}{N_t}\sum_{n=1}^{N_t}\|\mathbf{r}^n(\mathbf{V}\mathbf{q}^n(\boldsymbol{\mu}))\|_1, \tag{3.40}$$

where $\mathbf{q}^n(\boldsymbol{\mu})$ denotes the reduced coordinates for the parameter $\boldsymbol{\mu}$ at time iteration t^n. The greedy procedure is recalled in Algorithm 1.

Algorithm 1 Greedy sampling of the parameter space

Require: Residual function $\mathbf{r}(\cdot)$, tolerance for convergence ϵ, candidate parameter set $\mathcal{C} = \{\boldsymbol{\mu}_i\}_{i=1}^{N_c}$
Ensure: Dictionary $\mathbf{V}$

Randomly chose an initial sample parameter $\boldsymbol{\mu}_0 \in \mathcal{C}$ and compute the associated HDM solution $\{\mathbf{w}^n(\boldsymbol{\mu}_0)\}_{n=1}^{N_t}$
Construct an initial dictionary $\mathbf{V} = \{\mathbf{w}^n(\boldsymbol{\mu}_0)\}_{n=1}^{N_t}$
for $i_c = 1, \cdots, N_c$ **do**
    Solve for the ROM solution $\{\mathbf{q}^n(\boldsymbol{\mu}_{i_c})\}_{n=1}^{N_t}$ and evaluate the error indicator $\mathcal{E}(\boldsymbol{\mu}_{i_c})$
end for
$j = 1$
while $\max_{i_c=1,\cdots,N_c}\mathcal{E}(\boldsymbol{\mu}_{i_c}) > \epsilon$ **do**
    Select $\boldsymbol{\mu}_j = \operatorname{argmax}_{i_c=1,\cdots,N_c}\mathcal{E}(\boldsymbol{\mu}_{i_c})$
    Compute the associated HDM solution $\{\mathbf{w}^n(\boldsymbol{\mu}_j)\}_{n=1}^{N_t}$
    Update the dictionary $\mathbf{V} = \mathbf{V}\bigcup\{\mathbf{w}^n(\boldsymbol{\mu}_j)\}_{n=1}^{N_t}$
    for $i_c = 1, \cdots, N_c$ **do**
        Solve for the ROM solution $\{\mathbf{q}^n(\boldsymbol{\mu}_{i_c})\}_{n=1}^{N_t}$ and evaluate the error indicator $\mathcal{E}(\boldsymbol{\mu}_{i_c})$
    end for
    $j = j + 1$
end while

Remark 3.6.1. *In the case of a monotone scheme and convex hull minimization, the result of Proposition 3.5.1 shows that the error indicator (3.40) can be bounded by $\mathcal{J}(\mathcal{D})$. This can also be observed from the numerical experiments in Section 3.9, where the chosen greedy parameters in the dictionary are maximally separated.*

3.7 Potential difficulties using L^1-norm and algorithms

In this section, we are firstly illustrating the potential difficulties that one can encounter when using L^1-norm minimization and how to fix them. In the second part of this section, model reduction based on minimizing the residual in the L^1-norm is combined with the dictionary approach presented in Section 3.3 which leads to different algorithms used to solve the minimization problem.

3.7.1 Potential difficulties and procedures

As an alternative to Galerkin projection and residual minimization in the least-squares sense, a reduced system of equation is here obtained by minimizing, at each time step $n = 1, \cdots, N_t$, the L^1-norm of the residual vector as

$$\min_{\mathbf{q}\in\mathbb{R}^k} \|\mathbf{r}^n(\mathbf{V}\mathbf{q})\|_1 = \sum_{i=1}^{n} |r_i^n(\mathbf{V}\mathbf{q})|\,,\ n = 1, \cdots, N_t \tag{3.41}$$

or the projection on the convex envelop of the dictionary defined in (3.34). There are at least three difficulties associated with minimizing the L^1-norm. The first one is due to L^1 non-differentiability at zero. To circumvent this issue, the Huber function [71], defined as follows can be introduced:

$$\phi_M(x) = \begin{cases} x^2 & \text{if } |x| \leq M \\ M(2|x| - M) & \text{otherwise,} \end{cases} \tag{3.42}$$

Then, the sequence of reduced systems of equations based on the Huber function is

$$\min_{\mathbf{q}\in\mathbb{R}^k} \sum_{i=1}^{n} \phi_M\left(r_i^n(\mathbf{V}\mathbf{q})\right),\ n = 1, \cdots, N_t. \tag{3.43}$$

The Huber function ϕ_M behaves as a parabola close to $x = 0$ and as the L^1-norm for large values of x. It is continuously differentiable on $\mathbb{R}$ ($\phi_M \in C^1(\mathbb{R})$). It is also used in regressions as a loss function due to its non-sensitivity to outliers. In the present work, it will be used as a continuously differentiable alternative to the L^1-norm.

Figure 3.1 compares, in the scalar case, the L^2 and L^1-norms to the norm based on the Huber function for the particular case $M = 1$. Practical algorithm for solving the systems of equations (3.41) and (3.43), both in the case of linear and nonlinear residual functions are presented in the following section.

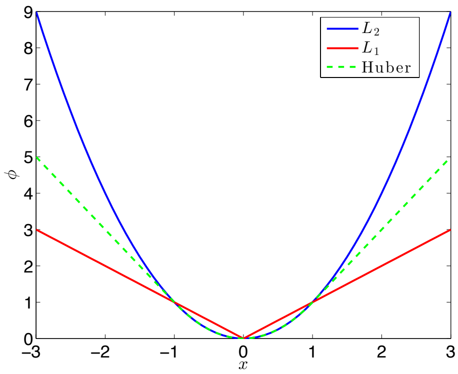


Figure 3.1: Comparison of the L^2, L^1 and Huber function $(M = 1)$ norms

A second difficulty is that the L^1-norm is not strictly convex, so that the uniqueness is not guaranteed. This difficulty is taken into account in the solution procedure by adding a strictly convex penalization term, for example a L^2-constraint (as in (3.27a) and (3.27b)).

A third potential issue using a dictionary, as opposed to a reduced basis, is the fact that the dictionary may be rank-deficient. One option to address this issue is to perform a Gram-Schmidt orthogonalization or a rank-revealing QR factorization. A drawback of that approach is that dictionary members are then linearly combined. Alternatively, a regularization term and a random perturbation is here added to the minimization functionals to ensure a system with full rank and a unique solution as follows:

- For L^1-norm minimization, the functional becomes

$$\min_{\mathbf{q}\in\mathbb{R}^k} \|\mathbf{r}^n(\mathbf{V}\mathbf{q})\|_1 + \nu\|\mathbf{q}\|_2^2 = \min_{\mathbf{q}} \sum_{i=1}^{N} |r_i^n(\mathbf{V}\mathbf{q})| + \nu \sum_{j=1}^{k} q_j^2, \;\; n = 1, \cdots, N_t, \;\; \nu > 0. \quad (3.44)$$

- For Huber function minimization, the functional becomes

$$\min_{\mathbf{q}\in\mathbb{R}^k} \sum_{i=1}^{N} \phi_M\left(r_i^n(\mathbf{V}\mathbf{q})\right) + \nu\|\mathbf{q}\|_2^2, \;\; n = 1, \cdots, N_t, \;\; \nu > 0. \quad (3.45)$$

Remark 3.7.1. *In numerical examples, we are using $\nu = 10^{-6}$ for L^1-norm minimization and $\nu = 10^{-8}$ for Huber function minimization. These values have been found to be robust throughout applications.*

3.7.2 Algorithms

A classical solution to minimizing a linear residual vector in the L^1-norm is by recasting the problem as a linear program (LP). More specifically, assuming that the residual is linear

$\mathbf{r}^n(\mathbf{Vq}) = \mathbf{A}^n\mathbf{Vq} + \mathbf{b}^n$ with $\mathbf{A}^n \in \mathbb{R}^{N\times N}$ and $\mathbf{b}^n \in \mathbb{R}^N$, a solution to (3.41) is given by the solution $\mathbf{q} \in \mathbb{R}^k$ of the LP

$$\min_{\mathbf{q},\mathbf{s},\mathbf{t}} \ \mathbf{1}^T(\mathbf{s}+\mathbf{t}) \tag{3.46}$$

$$\text{s.t. } \mathbf{A}^n\mathbf{Vq} + \mathbf{b}^n - \mathbf{s} + \mathbf{t} = \mathbf{0} \tag{3.47}$$

$$\mathbf{s} \geq \mathbf{0} \tag{3.48}$$

$$\mathbf{t} \geq \mathbf{0}. \tag{3.49}$$

Unfortunately, this LP involves $k + 2N$ variables and $3N$ constraints, including N equality constraints, rendering this approach intractable in the case of model reduction.

Alternatively, the L^1-norm minimization problem can be solved by Iteratively Reweighted Least Squares (IRLS) [45]. This approach proceeds iteratively by solving a sequence of weighted least-squares problem. An advantage of this approach is that its implementation can rely entirely on existing least-squares solvers. Furthermore, its complexity is similar to that of the L^2-norm minimization problem. The procedure is presented in Algorithm 2 in the case of a nonlinear residual vector. At each iteration l, a weighted least-squares problem is solved, where the weight depend on the current value of the residual vector $\mathbf{r}^l$ as follows: $\mathbf{\Theta}^l = \mathbf{diag}\left(|r_i^l|^{-\frac{1}{2}}\right)$.

Algorithm 2 L^1-norm minimization by Iteratively Reweighted Least-Squares

Require: Residual function $\mathbf{r}(\cdot)$ and associated Jacobian $\mathbf{J}(\cdot)$, reduced basis $\mathbf{V}$, initial guess $\mathbf{q}^0$, tolerance for convergence ϵ

Ensure: Solution $\mathbf{q}$

1: $l = 0$
2: Compute $\mathbf{r}^0 = \mathbf{r}(\mathbf{Vq}^0)$ and $\mathbf{Z}^0 = \mathbf{J}(\mathbf{Vq}^0)\mathbf{V}$
3: **while** $l = 0$ OR $\|\Delta\mathbf{q}^{l-1}|_1 > \epsilon(1 + \|\mathbf{q}^{l-1}\|_1)$ **do**
4: Compute the weights $\mathbf{\Theta}^l = \text{diag}\left(|r_i^l|^{-\frac{1}{2}}\right)$
5: Solve the weighted least-squares problem

$$\Delta\mathbf{q}^l = \underset{\mathbf{y}}{\text{argmin}}\, \|\mathbf{\Theta}^l\mathbf{Z}^l\mathbf{y} + \mathbf{\Theta}^l\mathbf{r}^l\|_2^2$$

6: $\mathbf{q}^{l+1} = \mathbf{q}^l + \Delta\mathbf{q}^l$
7: Compute $\mathbf{r}^{l+1} = \mathbf{r}(\mathbf{Vq}^{l+1})$ and $\mathbf{Z}^{l+1} = \mathbf{J}(\mathbf{Vq}^{l+1})\mathbf{V}$
8: $l = l + 1$
9: **end while**
10: $\mathbf{q} = \mathbf{q}^l$

Similarly, minimization of the Huber function can also be done by an IRLS procedure, as described in Algorithm 3. The procedure only differs from its L^1-norm counterpart by the choice of weights. In the present work, the following choice of weights is proposed for a given residual vector $\mathbf{r}^l$:

$$\mathbf{\Theta}^l = diag(\Theta_i^l), \tag{3.50}$$

where

$$\Theta_i^l = \begin{cases} 1 \text{ if } |r_i^l| < M \\ \frac{M}{\sqrt{|r_i^l|}} \text{ else} \end{cases} \tag{3.51}$$

and setting

$$M = \epsilon_2 \max\left(1, \max_i \left(|r_i^l|\right)\right). \tag{3.52}$$

The value $\epsilon_2 = 10^{-6}$ has been found to be a robust choice across different applications.

Algorithm 3 Huber function minimization by Iteratively Reweighted Least-Squares

Require: Residual function $\mathbf{r}(\cdot)$ and associated Jacobian $\mathbf{J}(\cdot)$, reduced basis $\mathbf{V}$, initial guess $\mathbf{q}^0$, tolerance for convergence ϵ
Ensure: Solution $\mathbf{q}$

$l = 0$
Compute $\mathbf{r}^0 = \mathbf{r}(\mathbf{V}\mathbf{q}^0)$ and $\mathbf{Z}^0 = \mathbf{J}(\mathbf{V}\mathbf{q}^0)\mathbf{V}$
while $l = 0$ OR $\|\Delta\mathbf{q}^{l-1}|_1 > \epsilon(1 + \|\mathbf{q}^{l-1}\|_1)$ **do**
  Compute the weights $\mathbf{\Theta}^l = \mathbf{diag}\left((|r_i^l| < M) + M|r_i^l|^{-\frac{1}{2}}(|r_i^l| \geq M)\right)$
  Let $M = \epsilon_2 \max(1, \max(|r_i^l|))$
  Solve the weighted least-squares problem

$$\Delta\mathbf{q}^l = \operatorname*{argmin}_{\mathbf{y}} \|\mathbf{\Theta}^l\mathbf{Z}^l\mathbf{y} + \mathbf{\Theta}^l\mathbf{r}^l\|_2^2$$

  $\mathbf{q}^{l+1} = \mathbf{q}^l + \Delta\mathbf{q}^l$
  Compute $\mathbf{r}^{l+1} = \mathbf{r}(\mathbf{V}\mathbf{q}^{l+1})$ and $\mathbf{Z}^{l+1} = \mathbf{J}(\mathbf{V}\mathbf{q}^{l+1})\mathbf{V}$
  $l = l + 1$
end while
$\mathbf{q} = \mathbf{q}^l$

3.8 Computational cost

Let us make some comments on the computational cost of the method.

The minimization procedure consists in looking for the minimum of functionals of the type (3.27) where all the degrees of freedom describing the solution appears. This is a challenge, since in general, the algorithms to solve this kind of minimization procedure are much more expensive than those of the least square type. In all the numerical experiments that will be presented in Section 3.9, we consider a slightly different approach namely, instead of using all the degrees of freedoms, we use only a small subset of them $\mathcal{I}$. Hence, instead of minimizing the total variation in (3.23), one can minimize the total variation over $\mathcal{I}$, i.e.

$$TV(\boldsymbol{r}) = \sum_{j\in\mathcal{I}} \left| \left[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\right]_j \right|. \tag{3.53}$$

In practice, because of the uniqueness issues listed above, instead of the functional (3.27b), we minimize:

$$J(\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})) = \sum_{j\in\mathcal{I}} \left| \left[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\right]_j \right| + \nu \sum_{j\in\mathcal{I}} \left(\mathbf{w}_j^{n+1}\right)^2, \tag{3.54}$$

where $\mathcal{I}$ is a small subset of the set of degrees of freedom defined as in (3.53). Of course the question is how to choose this set. Clearly, if $\mathcal{I}$ is equal to the full set of grid points, the solution is given by (3.23). We discuss how to choose $\mathcal{I}$ later in this paragraph, and focus first on the evaluation of $\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})$. We describe this in the steady two dimensional case, the extension to the unsteady case or the 1D case is quite straightforward. In our simulation we have chosen to use the scheme developed in [7, 8, 116] but this is not essential and what matters is that the stencil of the method is relatively compact.

3.8.1 Evaluation of $\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})$

The computational domain is covered by an unstructured mesh which elements K are triangles in 2D. The same method could deal with an hybrid mesh. Consider a vertex M_i in the computational mesh. We denote by $\mathcal{V}_i$ the set of vertices that are connected to M_i by an edge, and by $\mathcal{W}_i$ the set of points that are needed to evaluate $\big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_i$, see Figure 3.2. In the case of the scheme described bellow, the two sets coincide, and we also have that the set $\mathcal{W} = \mathcal{V}_i \cup \{M_i\}$ is the set of vertices contained in $\cup_{K, M_i \in K} K$. The full order model writes:

$$\big[\boldsymbol{r}(\mathbf{w}^n, \mathbf{w}^{n+1})\big]_i = |C_i| \frac{\mathbf{w}_i^{n+1} - \mathbf{w}_i^n}{\Delta t} + \sum_{K, M_i \in K} \Phi(\mathbf{w}_i^n, \mathbf{w}_j^n, \mathbf{w}_k^n).$$

Here, the vertices of K are M_i, M_j, M_k, and $|C_i|$ is the area of the dual control volume. Though the stencil of the method is reduced to $\mathcal{V}_i \cup \{M_i\}$, the method is second order in space.

In the case of the reduced order model, we proceed as follows: knowing $\mathbf{q}^n$, we evaluate the components of $V\mathbf{q}^n$ corresponding to the vertices of $\{M_i, i \in \mathcal{I}\} \cup \cup_{i \in \mathcal{I}} \mathcal{V}_i$. For a regular mesh, this amounts to $\approx 7|\mathcal{I}|$ evaluations. Then we evaluate $\mathbf{q}^{n+1}$ by the minimization of J defined in (3.54).

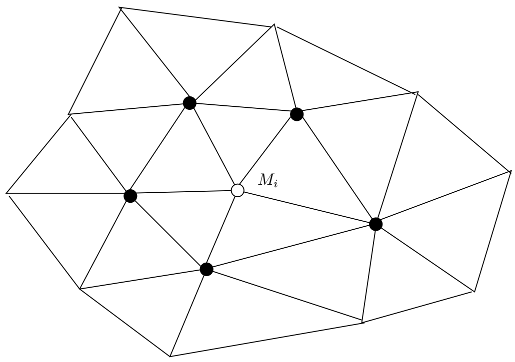


Figure 3.2: Representation of the set $\mathcal{V}_i$. The vertex M_i is indicated with a $\circ$, and the elements of $\mathcal{V}_i$ by $\bullet$.

3.8.2 Evaluation of $\mathcal{I}$

In the dictionary approach a lot of information is encoded and the degrees of freedom are linked if they are in the same cone of dependence. In the fluid dynamics case, and for a subsonic solution, the problem to solve is essentially elliptic hence, two different degrees of freedoms are linked together i.e, if one consider the grid points M_1 and M_2, the flow field at M_1 has some knowledge of what occurs at M_2. In the case of a transonic flow, the subsonic and supersonic pockets are somehow disconnected. Using this rational, one can select randomly points in the mesh, taking into account an *a priori* knowledge of the location of the supersonic pockets when they exist.

This hyper-reduction strategy is applied in all our numerical results. In 2D, we use only about 100 points in each case, out of 4510 points. Hence, only 2.2% of the total points is used. Several stencils are chosen and the presented results seem not to be sensitive to the choices. In addition, in the 1D case where the full grid can be used, we have not experienced any change in the solution.

3.9 Numerical applications

In the next numerical applications, we consider the parameter space $\mathcal{P}$ to be included in $\mathbb{R}^p$, with $p = 1$. As a consequence, we denote the parameters in this section by μ.

3.9.1 Unsteady Burgers' equation

We consider here the system

$$\frac{\partial w}{\partial t} + \text{ div } f(w) = 0 \tag{3.55}$$

defined on $\Omega = [0, \pi] \subset \mathbb{R}$, $t > 0$ with periodic boundary conditions, $f(u) = \frac{w^2}{2}$. The initial conditions are parametrized by

$$w_0(x; \mu) = \mu \left| \sin(2\, x) \right| + 0.1, \quad \mu \in [0.2, 0.7].$$

This initial condition is chosen such that a moving shock is generated in a finite time for $\mu \neq 0$. The shock moves with velocity is $\sigma_\mu = 0.5\mu + 0.1$. The PDE is discretized by an upwind first order scheme using a uniform mesh, resulting in a HDM of dimension $N = 10^2$, the CFL condition is 0.5, the number of iterations is $N_t = 200$ and the time step equals 0.0157.

The greedy sampling algorithm proposed in the Section 3.6 is firstly applied in an offline stage to construct a set of parameters $\mathcal{D}$ which is accurate in the parametric domain $\mathcal{P} = [0.2, 0.7]$. For that purpose, a set $\mathcal{C}$ containing $N_c = 11$ random training parameters is considered. We start with an initial element in the dictionary and then 6 greedy iterations are performed using a reduced order method based on L^1-norm minimization by LP and with the error indicator (3.40), resulting in a dictionary $\mathbf{V}$ with $k = 7$ members.

The values of $\mathcal{D}$ which represent the parameters selected by the greedy approach are reported in Figure 3.3 and the dictionary members of $\mathbf{V}$ are depicted in Figure 3.4. One can observe that the greedy algorithm selects in practice new samples that are maximally separated from the previously sampled dictionary members.

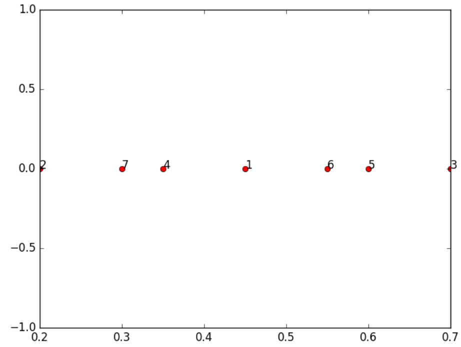


Figure 3.3: Parameters contained in $\mathcal{D}$, which are selected by the greedy algorithm for Burgers' equation

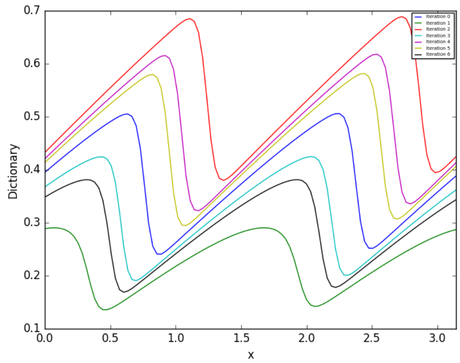
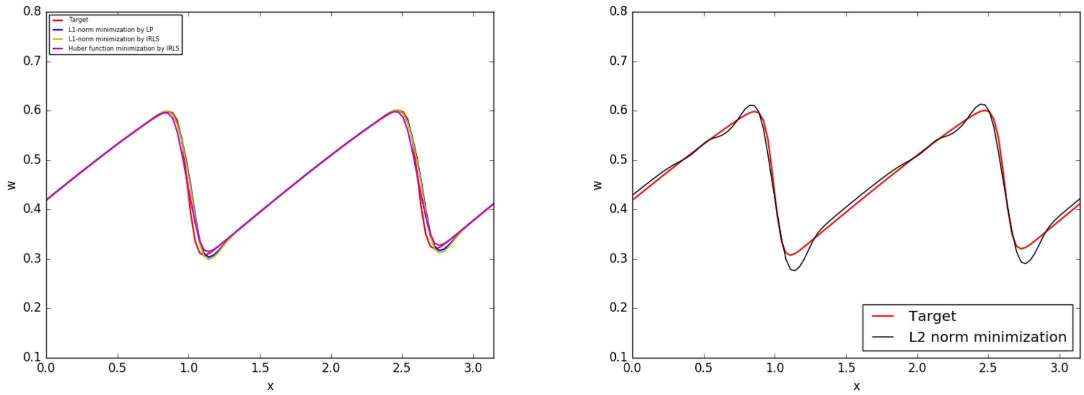


Figure 3.4: Members of the dictionary $\mathbf{V}$ at $t = \pi$ for Burgers' equation

For the online stage, two target parameters $\mu_1 = 0.575$ and $\mu_2 = 0.65$ are randomly selected and the dictionary approach based on the previously constructed 7 sampled is tested together with the following four model reduction approaches:

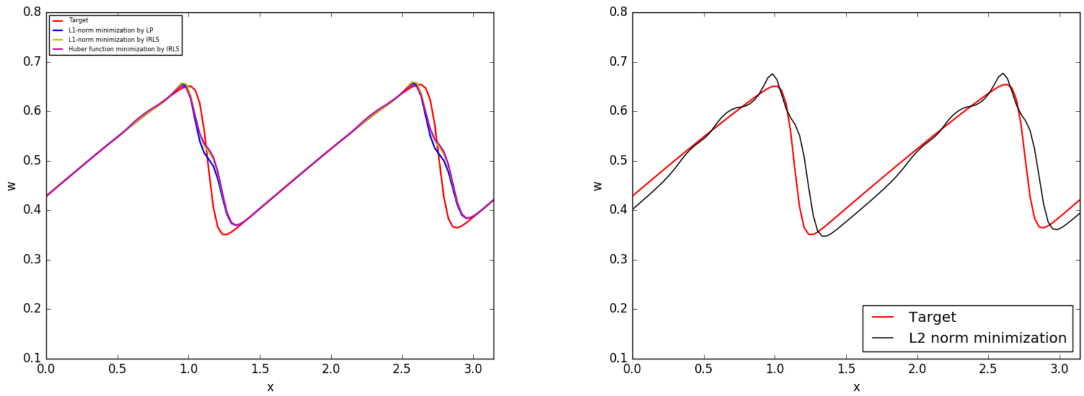
1. Minimization of the L^2-norm of the residual
2. Minimization of the L^1-norm of the residual by Linear Programming
3. Minimization of the L^1-norm of the residual by IRLS with tolerance $\epsilon = 10^{-8}$
4. Minimization of the Huber function applied to the residual by IRLS with tolerance $\epsilon = 10^{-8}$

The solutions obtained using each MOR approach are compared with the target solution in Figure 3.5 and Figure 3.6 at time $t = \pi$. Qualitatively, one can observe that the L^1-norm and Huber function-based approaches approximate the target solution the best by providing solutions with steep discontinuities. On the other hand, the L^2-norm minimization of the residual leads to a solution that presents undershoot and overshoots before and after the discontinuity, respectively.



(a) (b)

Figure 3.5: Comparison of all MOR approaches at $t = \pi$ for the target solution $\mu_1 = 0.575$



(a) (b)

Figure 3.6: Comparison of all MOR approaches at $t = \pi$ for the target solution $\mu_2 = 0.65$

The relative errors in L^1-norm between the true solution and each ROM solution

$$E_{rel}(\mu) = \frac{||w_{true}(\mu) - w_{ROM}(\mu)||_1}{||w_{true}(\mu)||_1} \tag{3.56}$$

computed for the target $\mu \in \mathcal{P}$ using different minimization procedures are reported in Table 3.1. One can observe that the approaches based on L^1-norm minimization (including the Huber function) lead to the smallest errors. In that same table, the CPU timings are reported. One can observe that the Linear Programming procedure is less computationally expensive that the IRLS approach and more than 3 times faster than the Huber approach. Nevertheless, even if the Huber function minimization approach is more expensive, it leads to much more accurate reduced solutions, as observed in Figure 3.5 and in Figure 3.6. These figures show the qualitative behavior of the L^1-norm types minimization in comparison with the quantitative results which are more or less in the same range for each norm considered, including L^2-norm.

We can use the RB method to approximate the solution of the problem (3.55) for a target parameter range from 0.2 to 0.7. In Figure 3.7 we report the error (3.56) as a function of the target parameter $\mu \in \mathcal{P}$. More precisely, we show the linear interpolation of the RB approximation error computed for 98 equidistant target values between 0.2 and 0.7. The vertical dashed lines are plotted in correspondence with the parameter values selected by the greedy Algorithm 1. It is obvious that the RB approximation error tends to vanish close to the values selected by the greedy algorithm. We also consider different mesh sizes, and performing the relative error in L^1-norm using L^1-norm minimization by LP and L^2-norm minimization, we can observe a slightly better order of convergence when using L^1-norm type minimizations than L^2-norm minimization (see Figure 3.8).

Table 3.1 Unsteady Burgers' equation: relative errors in L^1-norm and CPUs for solutions at time $t = \pi$, using different minimization techniques for a mesh with $N = 100$ points and the target $\mu_1 = 0.575$

	HDM	L^2-norm	L^1-norm (LP)	L^1-norm (IRLS)	Huber function (IRLS)
$E_{rel}(\mu)$	-	3.304e-6	1.510e-6	1.499e-6	1.229e-6
CPU timings (s)	7.391e-2	4.412e-4	6.108e-4	8.874e-4	1.959e-3

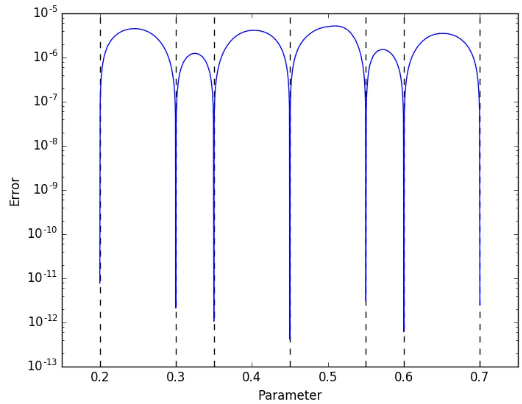


Figure 3.7: RB approximation error as a function of the target parameter in the range $[0.2, 0.7]$

Finally, the reduced coordinates associated with each ROM are reported in Figure 3.9. The L^1-norm and Huber function minimization types lead to sparse solutions whether L^2-norm minimization has only non-zero contributions from all dictionary members.

3.9.2 Euler equations

The one-dimensional Euler equations are considered on $\Omega = [0, 1]$

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho w \\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho w \\ \rho w^2 + p \\ w(E + p)\end{pmatrix} = 0, \tag{3.57}$$

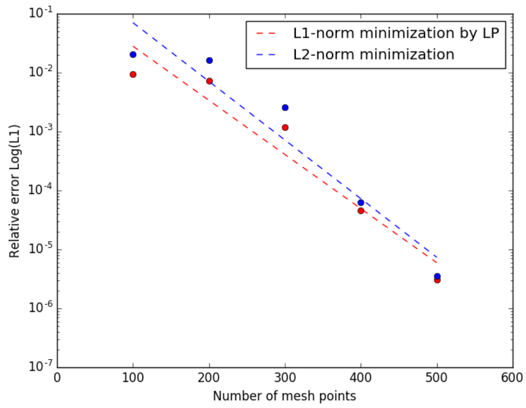


Figure 3.8: Convergence plots of the true relative errors using L^1-norm minimization by LP and L^2-norm minimization

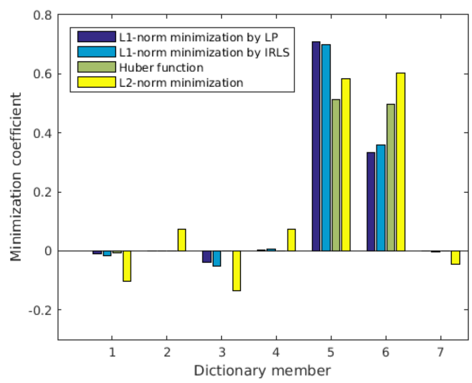


Figure 3.9: Reduced coordinates of the solutions for all MOR approaches at $\mu_1 = 0.575$

for which $\mathbf{W} = (\rho, \rho w, E)^T$ and the pressure is given by

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho w^2\right) \tag{3.58}$$

with $\gamma = 1.4$. For this problem, we have taken a second order finite volume scheme with MUSCL extrapolation on the characteristic variables and the limiter is minmod on all waves. The simulations are displayed at the final time $T_{fin} = 0.16$ and it is obtained with 100 grid points and a time step of 0.001. As in [5], we choose that the conservative initial conditions $\mathbf{W}_0(x;\mu), \mu \in [0,1]$ are constructed from the linear combination of the Sod and Lax shock tube problems. As suggested in [5], we have reconstructed the density, momentum and total energy independently, and here we are using the L^1-norm minimization by LP. The purpose of this subsection is to show the behavior of the reconstructed solution based on the number of elements in the dictionary $\mathbf{V}$ and for different targets.

The greedy sampling algorithm is applied to construct a set of parameters $\mathcal{D}$ that is accurate in the parametric domain $\mathcal{P} = [0.1, 0.8]$. For that purpose, a set $\mathcal{C}$ containing $N_c = 15$ training parameters randomly distributed are considered. An initial dictionary is considered and then

4 and 6, respectively, greedy iterations are performed resulting in a dictionary $\mathbf{V}$ with $k = 5$ and $k = 7$, respectively, members.

We start with the reconstruction of the solution for the target $\mu_1 = 0.3$ (see Figure 3.10) and then, in order to show that our L^1-minimization type can handle other parameters as a target, we are also illustrating the solution for $\mu_2 = 0.5$ (see Figure 3.11). One can observe that increasing just a little the number of elements in the dictionary, in our example only with two more reduced basis, will result in a more accurate reconstructed solution.

Remark 3.9.1. *Besides our effort to show by plotting the convergence, the sparsity, evaluating the errors and computing the CPUs, that L^1-norm minimization combined with a greedy algorithm improves the quality of the reduced solution, one can also compare these results with the ones obtained in [5] and a big improvement will be observed.*

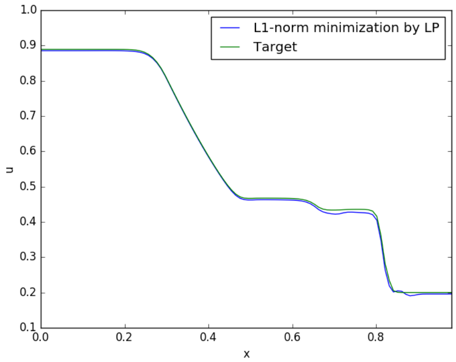
3.9.3 Nonlinear steady problems: a two dimensional example

The extension to multidimensions is straightforward. We have started from a code using a second order oscillation free method for solving the 2D Euler equations with subsonic boundary conditions on the outside boundary and no slip boundary conditions on the wing. A description of this method can be found, for example, in [1]. Note that this specific choice has no impact on the Reduced Order Model, since this algorithm is coded in Python on top of the CFD code which is called as a black box: any other CFD method would do the job. The used CFD mesh has 4510 grid points which corresponds to a total of 18040 unknowns. For this numerical experiment, the hyper-reduction is performed, using only a set of $\mathcal{I} = 100$ degrees of freedom, instead of $N = 4510$. Hence, only 2.2% of the total points is used.

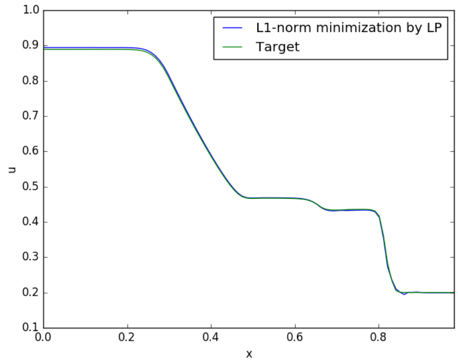
The minimization is done on the mass, momentum components and total energy: we introduce 4 sets of independent parameters which are the expansion coefficients for the dictionaries. In other words, we first run the CFD code to get a finite number of CFD solutions. Each solution is described as vector of state variables (ρ, m_x, m_y, E). From this we form four dictionaries, one for the density, two for the velocities and one for the total energy on which the reduced solutions are (independently) expanded. The minimization method is the straight L^1-minimization.

In order to illustrate the technique we have chosen a NACA012 profile with subsonic and transonic conditions. In the first case, the inflow mach number is $M_\infty = 0.65$ and the angle of attack (AoA) may change. The dictionary $\mathbf{V}$ is constructed by sampling the parameters $\mathcal{D} = \{-3.0°, -2.0°, -1.0°, 1.0°, 2.0°, 3.0°\}$ representing the AoA and the solution is sought for the predictive case $\mu = 0.0°$. We illustrate the elements in the dictionary (Figure 3.12) and then, using the L^1-norm minimization onto the convex hull of the dictionary, we obtain a ROM solution which is comparable with the exact numerical solution (Figure 3.13).

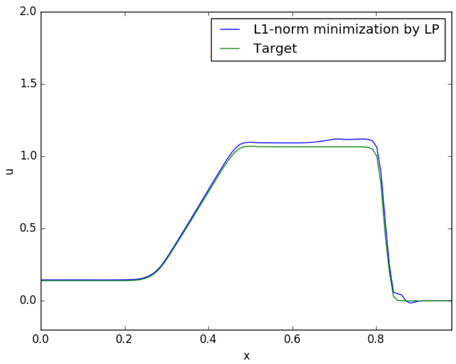
In the second case, we consider that a shock exists ($M_\infty = 0.85$) in order to illustrate that our method can also deal with this kind of problems. The dictionary $\mathbf{V}$ is constructed by sampling the parameters $\mathcal{D} = \{-3.0°, -2.0°, -1.2°, 1.0°, 2.0°, 3.0°\}$ representing the AoA. Note that the symmetry is intentionally broken in order to show that the symmetry do not have any influence on the result. The solution is sought for the predictive case $\mu = 0.0°$. As in the first case, we illustrate the elements in the dictionary $\mathbf{V}$, in order to show that



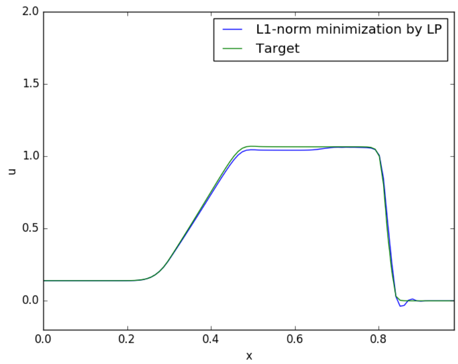
(a)



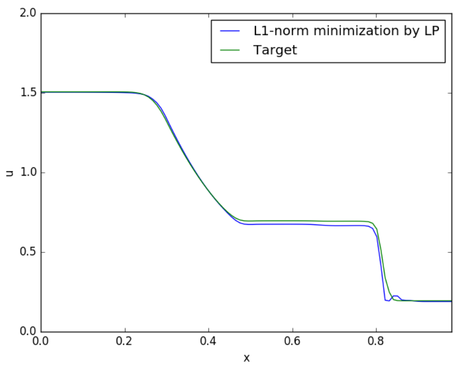
(b)



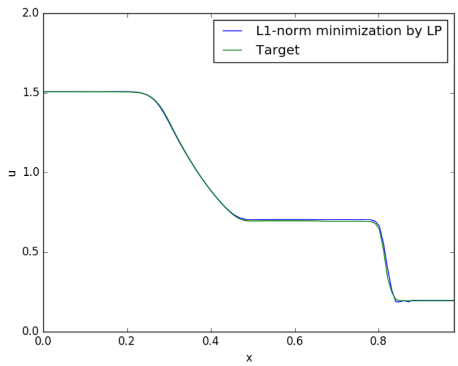
(c)



(d)



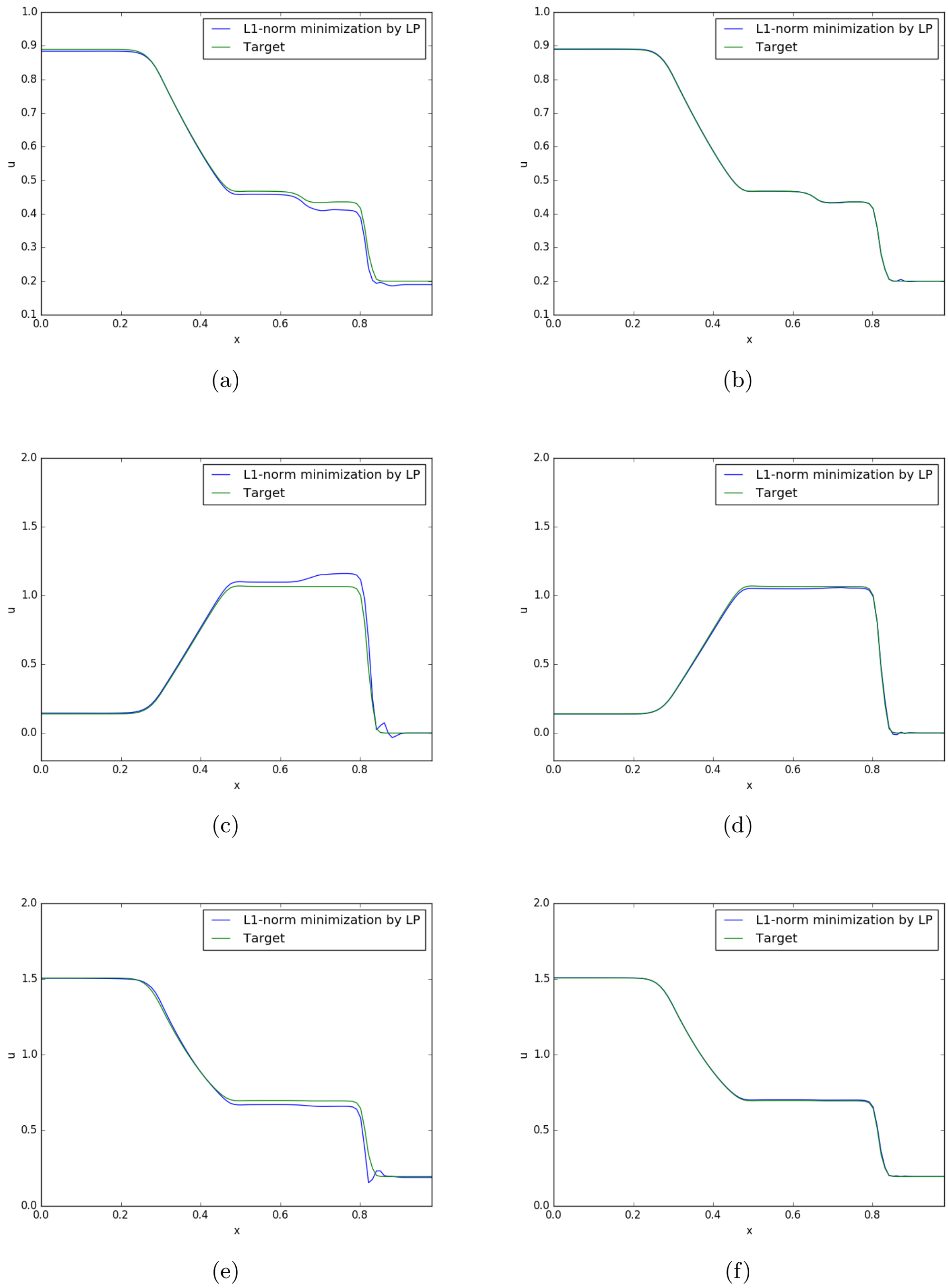
(e)



(f)

Figure 3.10: Reconstructed solution of density, momentum and total energy with 5 (left) and 7 (right) elements in the dictionary for the target $\mu_1 = 0.3$ at time $t = 0.16$

they are different (Figure 3.14). We obtain the following ROM solution using the L^1-norm minimization onto the convex hull of the dictionary (Figure 3.15). This one proves to be more robust: in this case we initialize by the projection on the dictionary of a uniform flow. We may



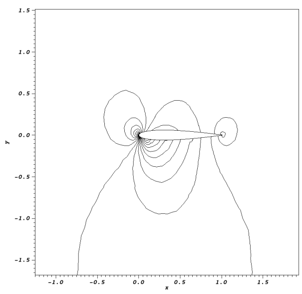
(a) (b)

(c) (d)

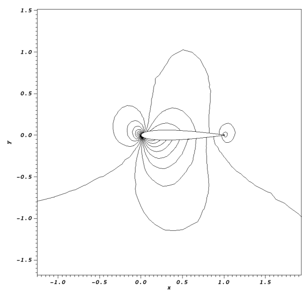
(e) (f)

Figure 3.11: Reconstructed solution of density, momentum and total energy with 5 (left) and 7 (right) elements in the dictionary for the target $\mu_2 = 0.5$ at time $t = 0.16$

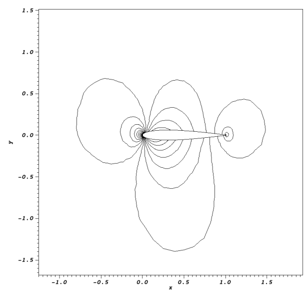
have positivity (of the density and/or the pressure) issues, and because of that, the projection on the convex hull revealed to be an efficient tool to control and avoid this issue. There is some discrepancy between the ROM solution and the exact numerical solution because the



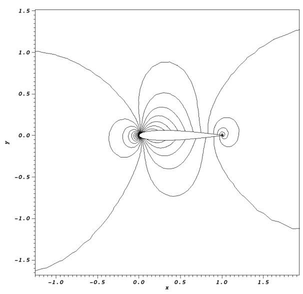
(a) AoA=−3.0°



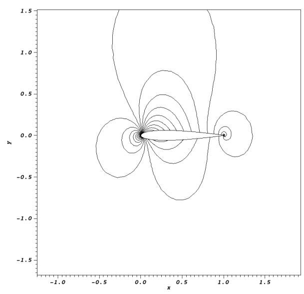
(b) AoA=−2.0°



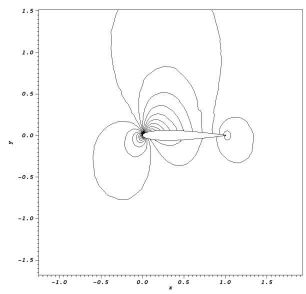
(c) AoA=−1.0°



(d) AoA=1.0°

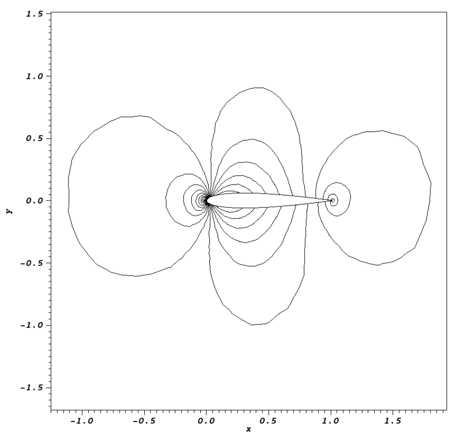


(e) AoA=2.0°

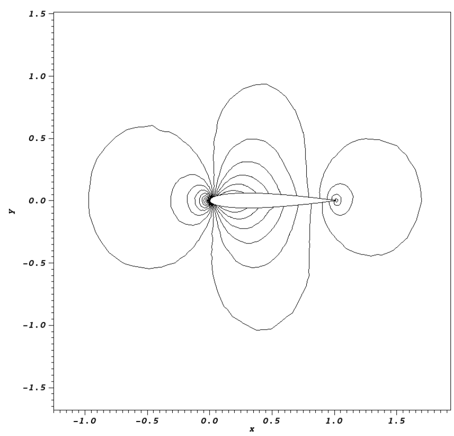


(f) AoA=3.0°

Figure 3.12: The components of the dictionary $\mathbf{V}$ constructed from the parameters $\mathcal{D} = \{-3.0°, -2.0°, -1.0°, 1.0°, 2.0°, 3.0°\}$ for Mach=0.65

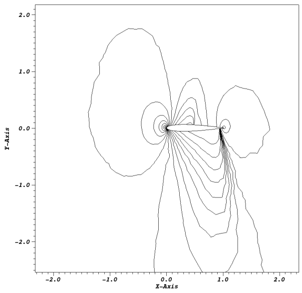


(a) ROM Solution

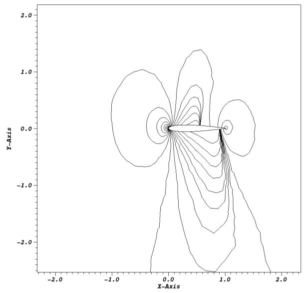


(b) Exact Numerical Solution

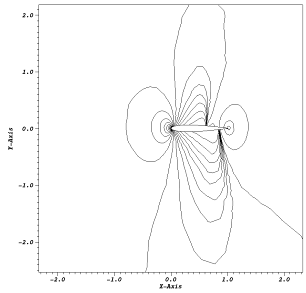
Figure 3.13: The ROM solution and the exact numerical solution for Mach= 0.65 and AoA=0.0°



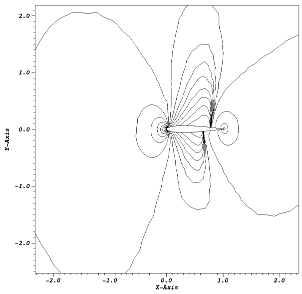
(a) AoA=$-3.0°$



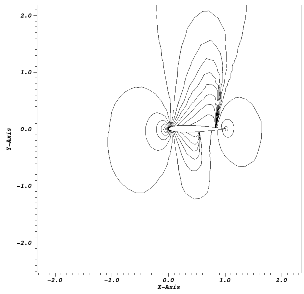
(b) AoA=$-2.0°$



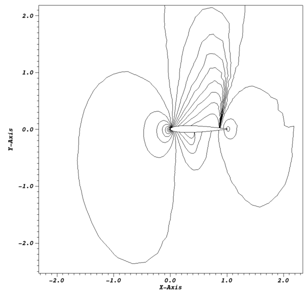
(c) AoA=$-1.2°$



(d) AoA=$1.0°$

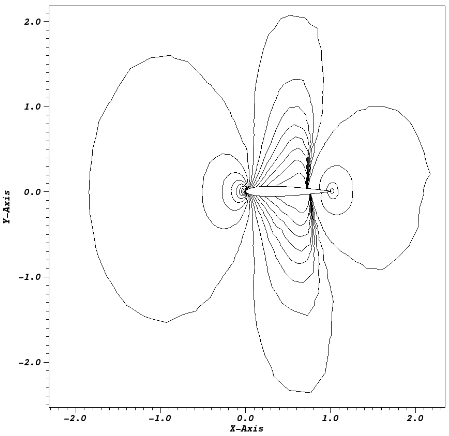


(e) AoA=$2.0°$

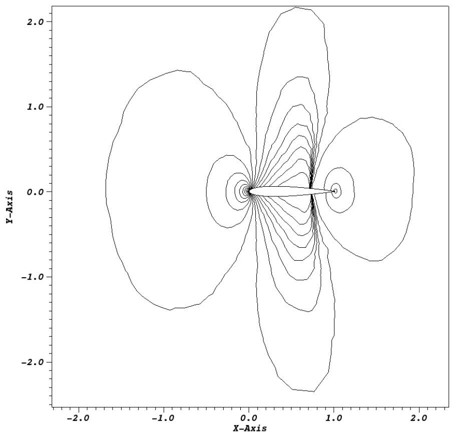


(f) AoA=$3.0°$

Figure 3.14: The components of the dictionary $\mathbf{V}$ constructed from the parameters $\mathcal{D} = \{-3.0°, -2.0°, -1.2°, 1.0°, 2.0°, 3.0°\}$ for Mach=0.85



(a) ROM Solution



(b) Exact Numerical Solution

Figure 3.15: The ROM solution and the exact numerical solution for Mach= 0.85 and AoA=$0.0°$

problem is very sensitive to the angle (this can be seen from the dictionary elements in Figure 3.14). As shown in Section 3.9.2, if we increase the number of elements in the dictionary, the ROM solution will be more similar to the exact numerical one, but in this case there is no simple strategy to obtain error bounds as in the scalar case.

Remark 3.9.2. *Treatment of moving boundary conditions is not as extensively studied for model reduction. Given that this problem includes shape changes, the wall boundary conditions are crucial as they are parametrized. In practice, a significant issue is that the solution snapshots emanating from different shapes lead to a reduced basis that does not satisfy all considered wall boundary conditions of the simulation performed online. A straightforward way to treat moving boundaries is to construct a separate set of bases for each possible boundary configuration [124]. This approach leads to an explosion of the required bases unless the boundary movement is restricted to a small subspace. For flows with periodic boundaries, and with inherent symmetries within the flow dynamics, it is possible remove uniform translation modes [119, 120]. In a restricted case when analyzing a single boundary in free flow, one can form the reduced basis in the frame of reference of the boundary as it is moved through various angles of flow attack [17]. Similarly, when a single moving boundary moves along a single dimension such as pistons within the engine cylinder, stretching and aligning of the flow basis can allow for a reduced model [52].*

Nevertheless, even if the boundary conditions are not very deep studied in this chapter, we can observe from the numerical experiments that, when using a dictionary approach, in the reduced solution, the velocity vector field is tangent to the wing, so the boundary conditions are satisfied (see Figure 3.16). Further details referring to the boundary condition issues are presented in Chapter 4.

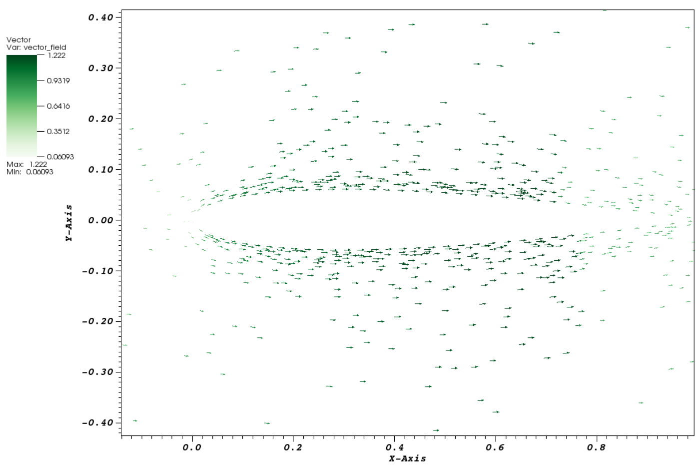


Figure 3.16: Velocity vector field of the ROM solution

Chapter 4: Model order reduction using Calibration

4.1 Introduction

In recent years, large-scale problems, often involving multiphysics, multiscale or stochastic behavior have become a particular focus of applied mathematics and engineering. A numerical treatment of such problems is usually very time-consuming and thus requires the development of efficient discretization schemes that are often realized in large parallel computing environments. In addition, these problems often need to be solved repeatedly for many varying parameters, introducing a curse of dimensionality when the solution is also viewed as a function of these parameters. These parameter dependent PDEs are called parametrized PDEs (PPDEs). The input parameters can characterize geometric features of the computational domain, some physical or material properties of the model at hand, initial and boundary conditions or source terms. Thus, fast reliable solutions to many queries PPDEs have many applications among which real time systems, optimization problems and optimal control.

For this class of problems, reduced order modeling (ROM) is a generic expression used to identify any approach aimed at replacing the high-fidelity problem by one featuring a much lower numerical complexity. Reduced basis (RB) methods enables to evaluate the solution of this latter problem, called reduced solution, for any new parameter instance, at a cost that is independent of the dimension of the original high-fidelity problem. They exploit the parametric dependence of the PDE solution by combining a handful of high-fidelity solutions computed for a set of parameter values. By this approach, a very large algebraic system is replaced by a much smaller one, whose dimension is related to the number of snapshots [69, 113].

An important notion in ROM is the solution manifold. Let $\mathcal{D}$ some parameter space on which we are studying the PPDE. For each parameter μ in $\mathcal{D}$, denote with $u(\mu)$ the high-fidelity solution. The solution manifold denoted by $\mathcal{M}_{\mathcal{D}}$ is defined as:

$$\mathcal{M}_{\mathcal{D}} := \{u(\mu),\ \mu \in \mathcal{D}\}.$$

Denote with X some normed linear space in which $\mathcal{M}_{\mathcal{D}}$ is embedded. The first question in a ROM context is how well $\mathcal{M}_{\mathcal{D}}$ can be approximated by a finite-dimensional subspace of prescribed dimension. The mathematical frame for this is linked to the notion of *Kolmogorov N-width* of solution manifold defined as:

$$d_N(\mathcal{M}_{\mathcal{D}}, X) = \inf_{E_N} \sup_{f \in \mathcal{M}_{\mathcal{D}}} \inf_{g \in E_N} \|f - g\|_X, \tag{4.1}$$

the first infimum being taken over all linear subspaces E_N of dimension N embedded in X.

In practice, one needs to build an algorithm to find subspaces close to the optimal ones given by the Kolmogorov n-width. The two most common strategies are the proper orthogonal decomposition (POD) [31, 32, 76, 82, 115, 130] and greedy approaches [111, 112, 122]. The latter leads to the construction of a "good" basis, "close" to the optimal one. More precisely, the resulting family of spaces, denoted by $\{X_N\}_N$, satisfies:

$$d_N(\mathcal{M}_\mathcal{D}, X) \approx \sup_{f \in \mathcal{M}_\mathcal{D}} \inf_{g \in X_N} \|f - g\|_X, \tag{4.2}$$

see [27, 47].

The optimality considered in the case of POD is slightly different. It focuses on minimizing the average error (parameter wise), in some norm. More precisely, we have the well known relation

$$\int_\mathcal{D} \|u(\mu) - \Pi_{POD} u(\mu)\|^2 \, d\mu = \sum_{i > N_{POD}} \lambda_i, \tag{4.3}$$

where Π_{POD} is the orthogonal projection onto the POD reduced space of dimension N_{POD} and the λ_i's are the eigenvalues of the associated correlation operator, in decreasing order. The faster the decay of the eigenvalues, the fewer modes are needed for a good (in average) reconstruction of the solution manifold.

It well known that the smallness of the N-width of the solution manifold of the PPDE of interest is a necessary condition before any ROM related method can be performed and there is an abundant litterature available when this property is satisfied. The other cases have recieved much less attention. We mention the results that we are aware of. They always rely on transformation of the solution manifold, to force the smallness of its N-width.

The first example that fall into this category is the Piola transform [91], which can map basis functions from a reference domain to each physical domain and which provides a better reduction than a simple change of variables. Piola transform is also used in the processing of the velocity field when the PDE is the Stokes or Navier Stokes problem and the parameter includes the geometry of the computational problem.

Another example is the freezing method [24, 106], which was later adapted in [34]. It focuses on handling convection dominated problems, and relies on the notion of calibration, a "preconditioning" step. It requires a family of (smooth) invertible mappings

$$\mathcal{F}_\mathcal{D} = \left\{ F : \overline{\Omega} \mapsto \overline{\Omega} \right\}, \tag{4.4}$$

in which there exist well chosen applications

$$\begin{array}{cccc} [0,T] & \times & \mathcal{D} & \to & \mathcal{F}_\mathcal{D} \\ (t,\mu) & & & \mapsto & F_{t;\mu} \end{array} \tag{4.5}$$

such that the corresponding preconditioned solution manifold, defined as:

$$\mathcal{M}_{\mathcal{F},\mathcal{D}} := \left\{ u(F_{t;\mu}^{-1}(\cdot), t; \mu),\ \mu \in \mathcal{D},\ t \in [0,T] \right\} \tag{4.6}$$

has a smaller Kolmogorov N-width than $\mathcal{M}_\mathcal{D}$. Behind this abstract formulation is the generalization of a simple idea. For periodic convection dominated problems, a well chosen translation reduces drastically the complexity of the problem. This idea was used in [34], where they provide numerical evidences that tend to show the viability of such methods for the development of computationally efficient schemes.

The objective of this chapter is to apply the calibration techniques developed in [34], to more realistic problems than the one dimensional Burgers equation. We have decided to focus on the steady two dimensional Euler equation around an airfoil. The precise setting will be discussed in Section 4.2. To motivate the need of calibration in this specific setting, we refer to the illustrative Figure 4.1. The colored lines are going through the barycenters of the mesh elements in which the gradient of the solution is the largest, for various pair of parameters: Mach number and angle of attacks (AoA). Each black line is a fitted line through the position of these barycenters. Because of the moving shock, the Kolmogorov N-width of the raw data

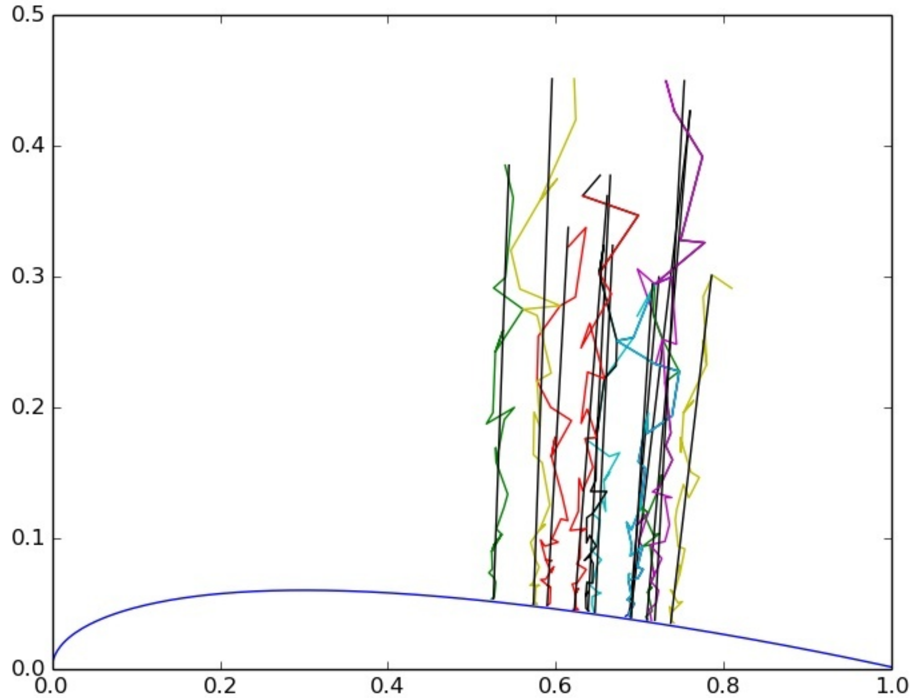


Figure 4.1: Position of the shock for various AoA and Mach numbers.
Colored lines: Barycenters of the cells in which the shock is located;
Black lines: Fitted lines through these barycenters

set $\mathcal{M}_{\mathcal{D}}$ will not have the good decay properties required for standard ROM. Thus, we need a preconditioning step, and we will propose an appropriate calibration.

In this chapter, we will follow the steps of [34], for this 2D hyperbolic problem. That is, we want to:

- calibrate the offline computed solution, to get a reduced basis as small as possible;
- have an online scheme that builds a "calibrated problem", making use of the calibrated reduced basis.

However, the novelties of this chapter in comparison with the work described in [34] are based on the increased difficulty and on the additional problems encountered when solving 2D parameter dependent hyperbolic conservation law problems as (4.7). Firstly, the shocks' position and shape might require more than one calibration parameters. Secondly, in this chapter, we are using subsonic boundary conditions on the outside boundary and no slip boundary conditions on the wing, which implies that we need a match with the exterior domain and as a result, the calibrated problem will not be only a translated version of the initial problem. The last difficulty which is taken into account in this chapter is related to the fact that standard CFD codes often imply numerical stabilization, which is not fitting in the reduced setting.

In the first section of this chapter, we completely describe the problem we want to solve. We then give details on the 'truth' scheme we are using. In the Section 4.3, we describe our choice of family of mappings $\mathcal{F}$, as well as one possible choice for $\mu \rightarrow F_\mu$. We use this to perform the 'offline phase'. We make sure that the calibration procedure leads to a better behaved solution manifold. In the Section 4.4, we propose a cheap 'online' algorithm. This is the central part of this chapter, as most related work simply perform the offline calibration, and do not propose any numerical scheme actually using the calibrated manifold $\mathcal{M}_{\mathcal{F},\mathcal{D}}$, see [72, 117, 138]. In the online phase, we propose a standard L^2-norm minimization algorithm and a L^1-norm extension, as was advised in [5]. In order to make the overall method computationally efficient, we describe how one could adapt hyper-reduction ideas [123]. The final section is devoted to numerical experiments. We present different mappings and we show the importance of the smoothness of the mappings in $\mathcal{F}$. We conclude this chapter by presenting some ideas that could be further investigated and implemented.

4.2 Problem setting

4.2.1 The 2 dimensional Euler equation

We are interested in the numerical approximation of the two dimensional Euler equations described in (4.7). We denote by Ω a domain around an airfoil which will be presented in the next section, $\mathbf{W}$ the state vector of conserved variables

$$\mathbf{W} = (\rho, \rho u, \rho v, E)^T$$

and $\mathbf{f} = (\mathbf{f}_x, \mathbf{f}_y)$ the flux is given by

$$\mathbf{f}_x(\mathbf{W}) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T$$
$$\mathbf{f}_y(\mathbf{W}) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T,$$

where ρ is the density, u and v are the components of the velocity, $E = \rho\epsilon + \frac{1}{2}\rho(u^2+v^2)$ is the total energy and ϵ is the specific internal energy. The system is closed by the equation of state relating the pressure p to the conserved variables:

$$p = (\gamma - 1)\big(E - \frac{1}{2}\rho(u^2+v^2)\big) = (\gamma-1)\rho\epsilon,$$

where the ratio of the specific heat γ is constant, with $\gamma = 1.4$ in our applications.

We are interested in the steady solutions. We will take them as the steady limit of the following evolution equation:

$$\begin{cases} \mathbf{W}_t + \operatorname{div} \mathbf{f}(\mathbf{W}) &= 0, \qquad t > 0, \mathbf{x} \in \Omega \\ \mathbf{W}(\mathbf{x}, 0) &= \mathbf{W}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega. \end{cases} \tag{4.7}$$

This problem is supplemented with boundary conditions which are specified in the next subsection.

We will take a quick glance at the fine computational method we are using, the Residual Distribution (RD) method, which is a second order oscillation free method. A complete description of this method for steady problems can be found, for example, in [2, 46].

4.2.2 Naca0012 test case

We have chosen to perform the calibration on the following well documented external flow test-case: the two-dimensional, inviscid, transonic flow past the NACA 0012 airfoil. The explicit form of the wing is given as:

$$y = w(x) := 0.6 \cdot \Big(0.2969 \cdot \sqrt{x} - 0.1260 \cdot x - 0.3516 \cdot x^2 + 0.2843 \cdot x^3 - 0.1015 \cdot x^4\Big), \text{ for } x \in [0, 1]. \tag{4.8}$$

We are using subsonic boundary conditions on the outside boundary and no slip boundary conditions on the wing. The latter is a Neumann type boundary condition, which imposes that the velocity of the fluid is tangent to the wing.

It is commonly known, that from a certain threshold of the Mach number, a shock appears. Both the position and the form of the shock depend on many parameters among which the Mach number and the angle of attack (AoA), i.e the inflow mean direction.

4.2.3 Residual distribution scheme

This short presentation of the RD scheme follows the lines of [46]. In order to approximate the solutions (4.7), we are using a conforming mesh with triangular elements. We will denote with T some generic element in the mesh, with $\mathcal{N}$ the number of elements in the mesh and by M a generic vertex. In the RD schemes, the solution of (4.7) is approximated at the vertices: the numerical approximation is represented by $(\mathbf{W}_i)_i$. From this, we construct a continuous interpolant $\mathbf{W}^h$ such that the function is linear on each triangle T and $\mathbf{W}^h(M_i) = \mathbf{W}_i$. The scheme also requires a continuous approximation of the flux $\mathbf{f}(\mathbf{W})$ over elements, which will be denoted $\mathbf{f}(\mathbf{W}^h)$.

Definition 4.2.1. *Let some current state $\mathbf{W}_i$, and $\mathbf{f}(\mathbf{W}^h)$ the corresponding continuous approximation of the flux.*

1. *$\forall T \in [1, \ldots, \mathcal{N}]$ compute the residual*

$$\Phi^T := \int_T div\ (\mathbf{f}(\mathbf{W}^h))d\mathbf{x} = \int_{\partial T} \mathbf{f}(\mathbf{W}^h) \cdot \vec{\mathbf{n}}\ d\tilde{\mathbf{x}}. \tag{4.9}$$

2. *$\forall T \in [1, \ldots, \mathcal{N}]$ distribute the functions of Φ^T to each node of T following the procedure in [1]. Denote by Φ_i^T the local nodal residual for the node $M_i \in T$. Thus, the RD construction will lead to:*

   – *Conservation relation*
$$\sum_{M_i \in T} \Phi_i^T = \Phi^T. \tag{4.10}$$

   – *If $\mathbf{W}^h$ is a piecewise linear interpolant of the exact smooth solutions of (4.7), then the techniques in [1] guarantees that*
$$\Phi_i^T = \mathcal{O}(h^3),$$
   for any vertex M_i and any triangle T, such that $M_i \in T$. In other words, it is shown that a converged RD scheme
$$\text{for all } M_i, \quad \sum_{T\ s.t\ M_i \in T} \Phi_i^T = 0 \tag{4.11}$$
   produces a second order accurate solution of the steady problem (4.7).

3. *We can consider the stabilized version of (4.11) by adding a SUPG stabilization term. Then, (4.11) becomes:*
$$\text{for all } M_i, \quad \sum_{T\ s.t\ M_i \in T} \Phi_i^{T,SUPG} = 0, \tag{4.12}$$
   where the stabilized split residuals take the form
$$\Phi_i^{T,SUPG} = \beta_i^T \Phi^T + h_T \int_T \Big(\nabla \mathbf{f} \cdot \nabla \rho_i^h\Big) \tau \Big(\nabla \mathbf{f} \cdot \nabla \mathbf{W}^h\Big) d\mathbf{x},$$
   with β_i^T being the distribution coefficient of node M_i , ρ^h a test function that vanishes on the inflow boundary, τ a scaling parameter depending on the mesh size and h_T a smoothness sensor, which assures that the additional stabilization term is only added in smooth regions of the solution [2, 3].

Note that as we are dealing with a system, the previous equality is to be understood in $\mathbb{R}^4$. The resolution uses an iterative process (pseudo time-stepping) to get to the solution $\{\mathbf{W}_i\}_i$.

This is a very general formulation and many classical schemes can be formulated within this framework. First, one can modify the way the residual of each triangle is distributed among nodes, that is, the choice of the β_i. For instance, distributing the residual evenly among nodes corresponds to a Lax-Friedriech type of scheme. One can achieve upwindng by taking into account the transport direction when distributing the residual. We have chosen a Lax-Friedriech type of scheme, with a SUPG stabilization (see [2] for more details). The consequences of these particular choices will be discussed in the online section.

The used fine CFD mesh has 4510 grid points which corresponds to a total of 18040 unknowns. Snapshots of the solution manifold can be visualized in Figure 4.2. We have identified a range of parameters, for which the shock position is sensitive to the change of Mach and AoA:

$$\mathcal{D} := \begin{cases} \text{Mach} & \in \quad [0.81, 0.83] \\ \text{AoA} & \in \quad [0.0^\circ, 3.0^\circ]. \end{cases}$$

The positions of the shocks for the sampled parameters in $\mathcal{D}$ are depicted in Figure 4.1. This problem has been already studied in Chapter 3 in the context of model reduction using L^1-norm minimization. It was shown in Subsection 3.9.3 that in the presence of shocks, discrepancies in the reduced solution are developed (see Figure 3.15(a)). This is actually the motivation of the work in this chapter.

In the rest of this chapter, we will denote u a generic component of the state vector $\mathbf{W}$. For instance, one component of the output of the CFD code for parameter μ will be denoted $u(\cdot;\mu)$. This choice of notation is not made to confuse the reader, but rather to match the standard notation in the ROM community, as already seen in Chapter 3.

4.3 Offline phase

As we will use a POD method to construct a reduced basis, we first need to select a moderate but representative snapshot set inside $\mathcal{M}_{\mathcal{D}}$. We have chosen the following set of cardinal 12:

$$\begin{array}{ll} \text{Mach} & \in \{0.81, 0.82, 0.83\} \\ \text{AoA} & \in \{0.0^\circ, 1.0^\circ, 2.0^\circ, 3.0^\circ\}. \end{array}$$

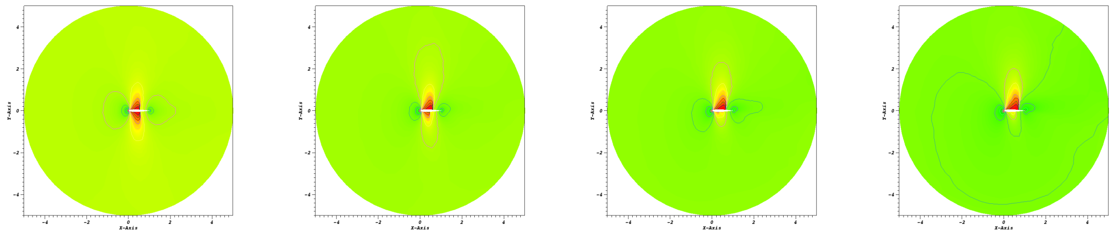
These snapshots are presented in Figure 4.2. We illustrate in Figure 4.3 a few basis resulting from the application of POD to this data set. One can observe that just as in the 1D Burgers' case, in order to take into account the variability of the shock positions and shapes, the reduced basis tend to oscillate. This behavior is even clearer when looking at the restriction of the POD basis at the wing (see Figure 4.4).

The first objective of this section is to propose a calibration procedure to mitigate this issue. For this, we construct in the next section a family of mappings $\mathcal{F}$ as well as an application $\mu \to F_\mu$.

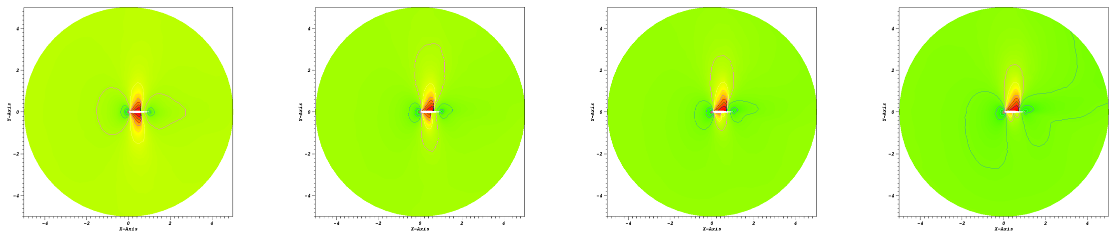
4.3.1 Preliminary remarks

As mentioned in the introduction, calibration starts with some a priori knowledge of the solution manifold. By analogy with the first dimensional Burgers' case, we choose the following calibration: let $\hat{\Omega}$ some reference domain and $\hat{x}_0$ some abscissa in $\hat{\Omega}$. Construct $\mathcal{F}$ a family of mappings from $\Omega \to \hat{\Omega}$ such that

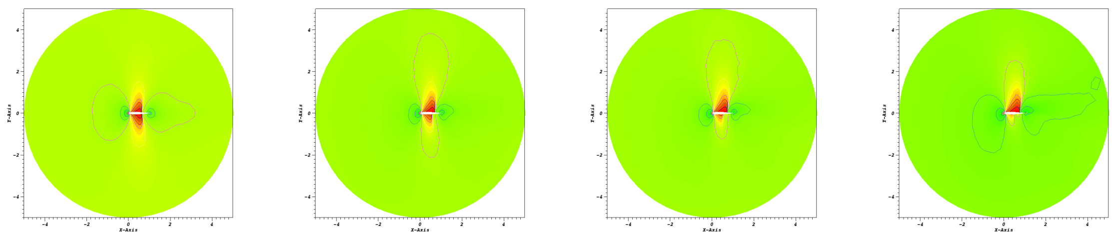
$$\forall \mu \in \mathcal{D},\ \exists F_\mu \in \mathcal{F},\ \left\{ (\hat{x}, \hat{y}) \in \hat{\Omega} \text{ s.t } u\left(F_\mu^{-1}(\cdot);\mu\right) \text{ is discontinuous } \right\} \subset \{(\hat{x}_0, \hat{y})\}$$



(a) Mach=0.81,AoA=0.0° (b) Mach=0.81,AoA=1.0° (c) Mach=0.81,AoA=2.0° (d) Mach=0.81,AoA=3.0°



(e) Mach=0.82,AoA=0.0° (f) Mach=0.82,AoA=1.0° (g) Mach=0.82,AoA=2.0° (h) Mach=0.82,AoA=3.0°



(i) Mach=0.83,AoA=0.0° (j) Mach=0.83,AoA=1.0° (k) Mach=0.83,AoA=2.0° (l) Mach=0.83,AoA=3.0°

Figure 4.2: The solutions of the problem (4.7) for AoA=$\{0.0°, 1.0°, 2.0°, 3.0°\}$ and Mach=$\{0.81, 0.82, 0.83\ \}$

To put it in other words, with this choice of calibration, the solutions in the calibrated manifold

$$\mathcal{M}_{\mathcal{F},\mathcal{D}} := \left\{ u\left(F_\mu^{-1}(\cdot);\mu\right),\ \mu \in \mathcal{D} \right\}$$

have vertical shocks, at position $\hat{x}_0$. Again, using the analogy with the one dimensional Burgers' case [34], we expect that the POD method applied to the calibrated manifold to represent better the shape of the solutions and to not try to catch the moving discontinuity.

How do we achieve this calibration? The first task is to locate the position of the shock. We have chosen the following simple strategy: first, find the boundary element (on the wing) where the quantity of interest has the highest gradient. Then, look at the neighboring elements and pick the one with the highest gradient. Iterate until the end of the shock (i.e some condition on the gradient) or until one reaches some predefined distance to the wing. One can use other methods in order to locate more precisely the shock. For instance, in [126], they use ENO related ideas to locate the inner-cell position of the shock.

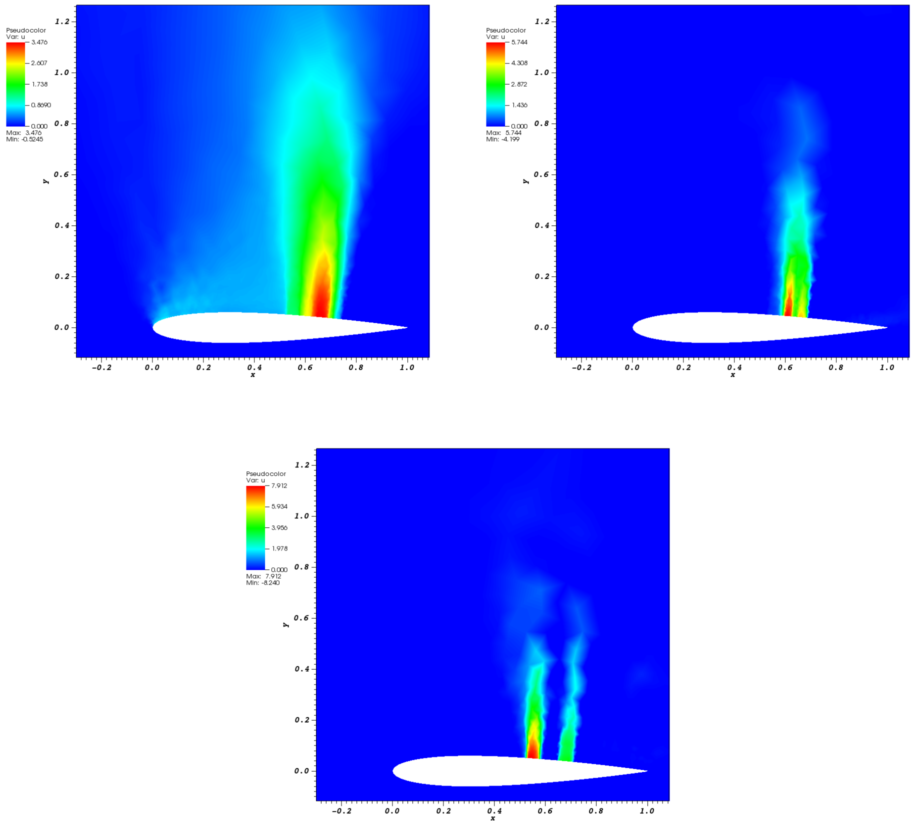


Figure 4.3: 1st, 3th and 5th POD basis at the wing in the uncalibrated case for the full domain Ω

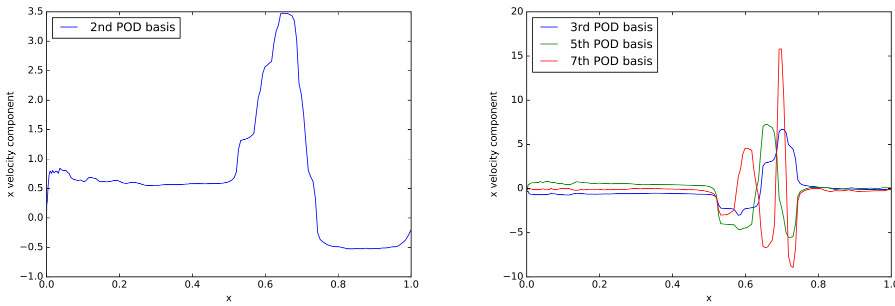


Figure 4.4: The x velocity component at the wing in the uncalibrated case : a few POD basis

We denote as $x = s(y;\mu), \mu \in \mathcal{D}$ the true shape of the shock and we will make the following assumption:

$$\exists k \text{ small }, \ \forall \mu \in \mathcal{D}, \exists P_\mu \in \mathcal{P}_k(\mathbb{R}), \ s(y;\mu) = \mathcal{P}_\mu(y). \tag{4.13}$$

That is, the shock can be represented by a low order polynomial. All numerical experiments presented in this chapter have been done using a first order polynomial:

$$P_\mu(y) = a_0(\mu) + a_1(\mu)\ y. \tag{4.14}$$

In Figure 4.1, the colored lines are the barycenters of the control volumes with the highest gradient. In black, is the fitted polynomial, characterized by two parameters, $a_0(\mu)$ and $a_1(\mu)$.

Second step now, we need to construct the family of mappings $\mathcal{F}$. The global picture is presented in Figure 4.5. We decompose the physical domain Ω into three subdomains:

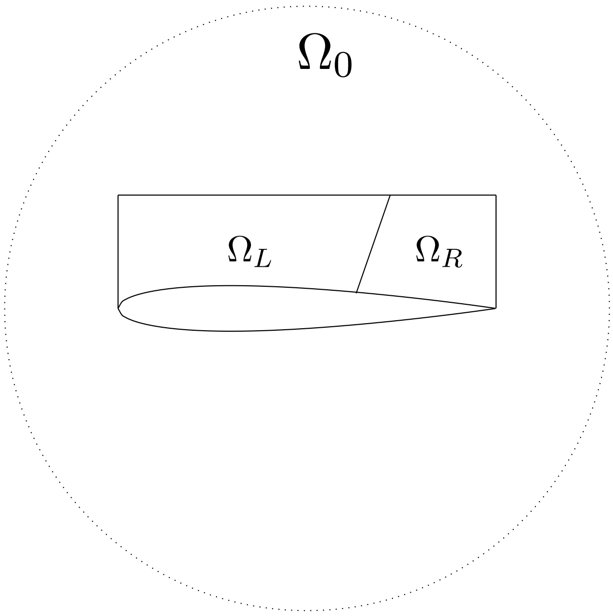


Figure 4.5: Physical domain Ω

- Ω_0, where we will use the identity mapping
- Ω_L and Ω_R, where we will perform the calibration.

We have chosen to use a Gordon-Hall (G-H) type of mapping [59], which will be presented in details in Section 4.3.2. Its properties have been studied in [92]. Different examples in fluid dynamics have been numerically studied in [91]. There are multiple reasons for this choice. First, its simplicity and flexibility are very important for the offline part. Second, for the online phase, its computational cost. In the rest of this section we will describe the application of the Gordon-Hall algorithm onto Ω_L. In a similar way, one can apply it also on the right subdomain Ω_R.

The reference domain has to be a rectangle in the original G-H algorithm. This fits in our framework, as we want the calibrated shock to be a vertical line. The situation is depicted in Figure 4.6, where we have plotted one possible instance of $F_\mu^{-1}(\hat{\Omega})$. Contrary to the most examples using Gordon-Hall method in the literature, our domain of interest is embedded in a bigger domain. The mapping thus needs to be (at least) continuous on $\partial\Omega_L$, and $\partial\Omega_R$. More precisely, we need

$$\begin{aligned}(x_1, y_1) &= (\hat{x}_1, \hat{y}_1)\\(x_3, y_1) &= (\hat{x}_3, \hat{y}_1)\\(x_3, y_2) &= (\hat{x}_3, \hat{y}_2)\\(x_1, y_2) &= (\hat{x}_1, \hat{y}_2).\end{aligned}$$

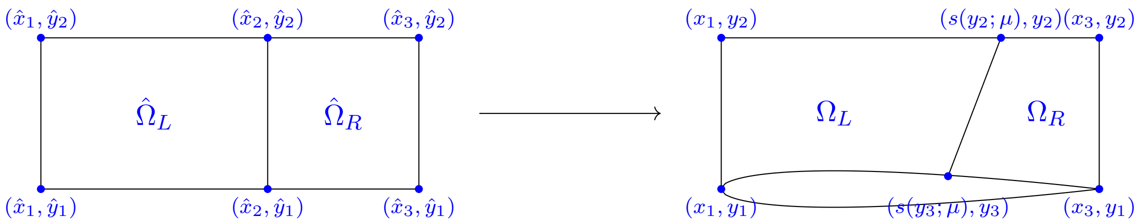


Figure 4.6: The reference domain $\hat{\Omega}$, and one possible instance $\Omega(\mu) := F_\mu^{-1}(\hat{\Omega})$

4.3.2 The actual G-H method

The G-H method is conceptually easy to understand. We denote with Γ_i the edges of Ω_L. We choose a clockwise numbering, starting from the left boundary. Their counterparts on $\hat{\Omega}_L$ are denoted $\hat{\Gamma}_i$. The steps are the following:

- map each edge of $\hat{\Omega}_L$ onto its counterpart on Ω_L. That is, define f_μ such that :

$$\forall i,\ f_\mu|_{\hat{\Gamma}_i} = \Gamma_i$$

- define the weights functions ϕ_i:

$$\begin{array}{ccc}\hat{\Omega}_L & \to & [0,1]\\(\hat{x}, \hat{y}) & \mapsto & \phi_i\end{array}$$

  satisfying the following necessary conditions :

$$\forall i \in [1, \ldots, 4],\ \begin{cases}\phi_i + \phi_{i+2} = 1\\ \phi_i|_{\hat{\Gamma}_i} = 1\end{cases}$$

  These functions represent the relative positioning between the opposing edges.

- define the projection functions π_i;

$$\begin{array}{ccc}\hat{\Omega}_L & \to & [0,1]\\(\hat{x}, \hat{y}) & \mapsto & \pi_i\end{array}$$

satisfying the following necessary condition :

$$\forall i \in [1, \ldots, 4], \begin{cases} \pi_i|_{\hat{\Gamma}_{i+1}} & = & 1 \\ \pi_i|_{\hat{\Gamma}_{i-1}} & = & 0 \\ \pi_i|_{\hat{\Gamma}_i} & \in & [0,1]. \end{cases}$$

These functions define a new coordinate system in $\hat{\Omega}_L$.

- for any point $(\hat{x}, \hat{y})$ on $\hat{\Omega}_L$, compute the projection onto each edge $\pi_i(\hat{x}, \hat{y})$. Then, use a weighted combination of the $f_\mu(\pi_i(\hat{x}, \hat{y}))$, where the weights are given by $\phi_i(\hat{x}, \hat{y})$'s.

Remark 4.3.1. *The conditions on the sets $\{\phi_i\}$ and $\{\pi_i\}$ stated above are necessary conditions. We have no explicit sufficient conditions to ensure the bijectivity of the G-H mapping.*

As a first easy step, we have chosen to linearly stretch/shrink the domain. That is, we choose the following parametrization of the edges Γ_i :

$$\begin{aligned} & f_\mu|_{\hat{\Gamma}_1} : (\hat{x}_1, \hat{y}) \to (\hat{x}_1, \hat{y}) \\ & f_\mu|_{\hat{\Gamma}_2} : (\hat{x}, \hat{y}_2) \to (\hat{x}_1 + \hat{x} \cdot (s(\hat{y}_2; \mu) - \hat{x}_1), \hat{y}_2) \\ & f_\mu|_{\hat{\Gamma}_3} : (\hat{x}_2, \hat{y}) \to (s(\hat{y}_2 + \hat{y} \cdot (y_3 - \hat{y}_2); \mu), \hat{y}_2 + \hat{y} \cdot (y_3 - \hat{y}_2)) \\ & f_\mu|_{\hat{\Gamma}_4} : (\hat{x}, \hat{y}_1) \to (s(y_3; \mu) + \hat{x} \cdot (\hat{x}_1 - s(y_3; \mu)), w(s(y_3; \mu) + \hat{x} \cdot (\hat{x}_1 - s(y_3; \mu))), \end{aligned} \tag{4.15}$$

where w is the shape of the wing defined in (4.8) and s is the shape of the shock given in (4.13). For example, the left edge $\hat{\Gamma}_1$ of the reference domain $\hat{\Omega}_L$, can be set to

$$\left\{ (\hat{x}, \hat{y}) \in \hat{\Omega}, \quad \text{s.t } \hat{y} \in [\hat{y}_1, \hat{y}_2] \text{ and } \hat{x} = \hat{x}_1 \right\}.$$

The vector valued function $f_\mu|_{\hat{\Gamma}_1}$ chosen above is only one possible parametrization of Γ_1.

Remark 4.3.2. *One can deduce from* (4.15) *that*

$$w(s(y_3; \mu)) = y_3 \quad \textit{and} \quad w(\hat{x}_1; \mu) = \hat{y}_1.$$

In this section, we consider the same weights and the same projection functions as in the original G-H formulation:

$$\phi_1(\hat{x}, \hat{y}) = \frac{\hat{y} - \hat{y}_1}{\hat{y}_2 - \hat{y}_1} \quad \phi_3(\hat{x}, \hat{y}) = 1 - \frac{\hat{y} - \hat{y}_1}{\hat{y}_2 - \hat{y}_1}$$

$$\phi_2(\hat{x}, \hat{y}) = \frac{\hat{x} - \hat{x}_1}{\hat{x}_2 - \hat{x}_1} \quad \phi_4(\hat{x}, \hat{y}) = 1 - \frac{\hat{x} - \hat{x}_1}{\hat{x}_2 - \hat{x}_1}.$$

and

$$\pi_1(\hat{x}, \hat{y}) = \frac{\hat{y} - \hat{y}_1}{\hat{y}_2 - \hat{y}_1} \quad \pi_3(\hat{x}, \hat{y}) = \frac{\hat{y} - \hat{y}_2}{\hat{y}_3 - \hat{y}_2}$$

$$\pi_2(\hat{x}, \hat{y}) = \frac{\hat{x} - \hat{x}_2}{\hat{x}_3 - \hat{x}_2} \quad \pi_4(\hat{x}, \hat{y}) = \frac{\hat{x}_3 - \hat{x}}{\hat{x}_3 - \hat{x}_1}.$$

The standard Gordon-Hall mapping is given by :

$$\begin{aligned} GH(\hat{x}, \hat{y}; \mu) = &\phi_1(\hat{x}, \hat{y}) \cdot f_\mu(\hat{x}_1, \hat{y}) + \phi_2(\hat{x}, \hat{y}) \cdot f_\mu(\hat{x}, \hat{y}_2) \\ &+ \phi_3(\hat{x}, \hat{y}) \cdot f_\mu(\hat{x}_2, \hat{y}) + \phi_4(\hat{x}, \hat{y}) \cdot f_\mu(\hat{x}, \hat{y}_1) \\ &- \sum_{i=1}^{4} \phi_i(\hat{x}, \hat{y}) \cdot \phi_{i+1}(\hat{x}, \hat{y}) \cdot f_{i;\mu}, \end{aligned} \tag{4.16}$$

where $f_{i;\mu}$ is the value of f_μ in the corner between Γ_i and Γ_{i+1}. Here, we have

$$\begin{aligned} f_{1;\mu} = (x_1, y_1), \quad f_{2;\mu} = (x_1, y_2) \\ f_{3;\mu} = (x_3, y_2), \quad f_{4;\mu} = (x_3, y_1). \end{aligned}$$

We will use, in the course of this chapter, the following notation :

$$\begin{array}{lcl} \mathbb{R}^2 & \to & \mathcal{F} \\ (a_0, a_1) & \mapsto & GH(\cdot; a_0, a_1) \end{array} \tag{4.17}$$

This application takes as argument a shock position (where the pair (a_0, a_1) represents the coefficients of the first order polynomial defined in (4.14)), and returns the corresponding Gordon-Hall mapping GH in $\mathcal{F}$.

Remark 4.3.3. *It is important to know that the π's, the ϕ's and f_μ can be chosen independently from each other. This will be made clearer in Section 4.7.2 when we try to improve the classical Gordon-Hall method 4.16.*

It is clear that this mapping suffers from major drawbacks :

- is continuous at the boundary, but has discontinuous derivatives;
- linearly stretches/shrinks the domain; this is not the best choice to diminish the Kolmogorov N-width;
- in x_1 and x_3, the boundary $\partial\hat{\Omega}$ is not C^1.

These issues will be fixed in the numerical section 4.7.2. Anyway, they are not a problem for the offline section. Thus, for simplicity, we will illustrate the usefulness of calibration using this rough mapping. We have computed separate POD basis on $\hat{\Omega}_L$ and $\hat{\Omega}_R$. We present in Figure 4.7 the counterpart of Figure 4.4, that is, the x component of the velocity on the left part of the wing. As one can see, using calibration we got rid of oscillations. We present in Figure 4.8 the first, third and fifth POD basis in the calibrated case, as a counterpart of Figure 4.3. As expected, the calibrated POD captures most of the information in the first 4 basis. The 5th basis contains only numerical noise. The first objective of this chapter, to construct a better behaved solution manifold has been solved.

We present in the next section a reduced scheme with a computational complexity independent of the size of the truth problem, which is based on the calibrated basis that we have just constructed.

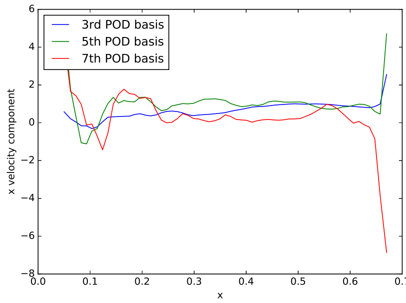


Figure 4.7: The x velocity component at the wing in the calibrated case : a few POD basis

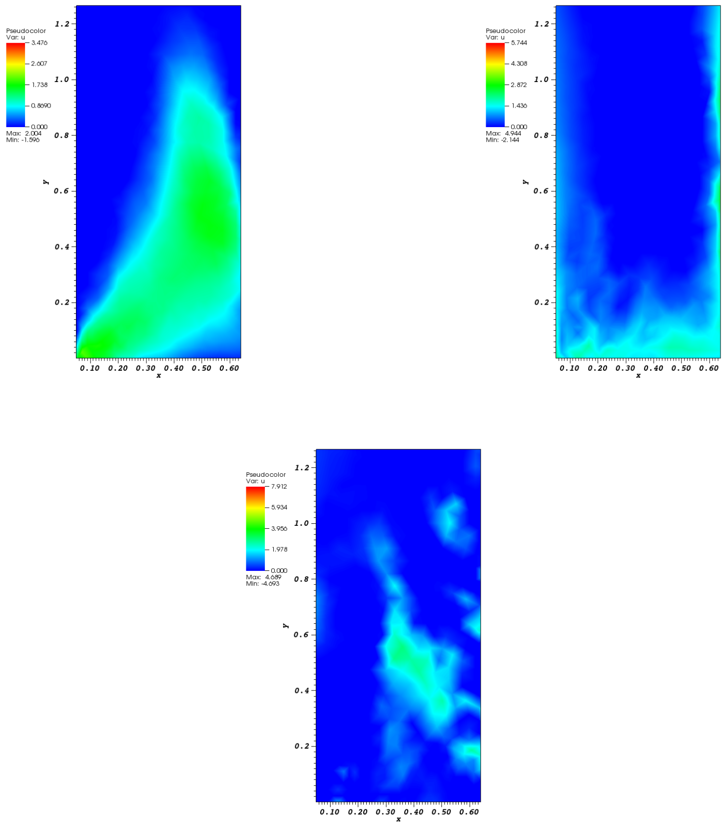


Figure 4.8: 1st, 3th and 5th POD basis in the calibrated case for the left subdomain

4.4 Online phase

For the remaining of this section, we drop out the μ dependency, as we are focused on reducing only one particular simulation. We will use again the μ dependency in the offline/online decomposition section. Also, we use the following notation:

- t^n denotes the discrete pseudo time;
- F_n is the mapping chosen at time step n. It maps Ω onto $\hat{\Omega}$. The inverse mapping will be denoted F_n^{-1};
- $\{\phi_i\}$ is some reduced basis on the reference mesh, of cardinal N^{red}. The one constructed in Section 4.3.

We denote by u^n the solution at pseudo time step t^n and $\hat{u}^n$ it's counterpart on the reference mesh. That is, we have

$$\hat{u}^n = u^n \circ F_n^{-1} \text{ on } \hat{\Omega}.$$

We present three different methods with increasing difficulty:

- The easiest method we can think of is the following:
  - suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - map this reduced solution onto the real mesh, using F_n;
  - use the CFD code, on Ω, using $\hat{u}^n \circ F_n$ as initial condition, to get u^{n+1};
  - map u^{n+1} back onto $\hat{\Omega}$. This implies finding a "good" (in some sense) mapping F_{n+1}, and the corresponding reduced coordinates.
- The second method is smarter, and more in the spirit of what has been done in [34]:
  - just as for the first method, suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - use a CFD code on $\hat{\Omega}$ using $\hat{u}^n$ as initial condition. This implies of course the modification of flux and boundary conditions to make this "non physical problem" equivalent to the initial one. Denote the output by $\tilde{u}^{n+1}$. Then, by construction, we have:
  $$\tilde{u}^{n+1} \approx u(\cdot, t^{n+1}) \circ F_n^{-1};$$
  - deduce a new "relative" mapping: $F_{n+1} \circ F_n^{-1}$ best suited to represent $\tilde{u}^{n+1}$. From this, compute a better calibrated solution $\hat{u}^{n+1}$ and the corresponding mapping F_{n+1} such that
  $$u(\cdot, t^{n+1}) \approx \hat{u}^{n+1} \circ F_{n+1};$$
- The third method is the ultimate goal of all reduced basis methods. This is based on constructing a self sufficient reduced scheme i.e the CFD code to be a black box. Reducing non trivial fluid simulations is known as being a very challenging problem because standard CFD codes often imply numerical stabilization, which is not fitting in the reduced setting. Then, the self sufficient scheme will necessarily rely on some new ingredients.

  In our opinion, two paths can be taken to actually build a self sufficient reduced scheme:

- Since we are working with reduced basis approximations of the parameter-dependent 2D Euler equation (4.7), we can stabilize the RB problem independent of the stabilization operated on the high-fidelity approximation, provided that a set of stable RB functions have been computed. This argument has been studied in [96] and is based on the transformation of the basis functions into modal basis, then on the addition of a vanishing viscosity term over the high RB modes, and on a rectification stage, to further enhance the accuracy of the RB approximation. In our case, the fine scheme is using the advanced Residual Distributed scheme described in Section 4.2 with SUPG type stabilization. The goal would be to use a simple scheme, say a raw Lax-Friedrichs method, which has no local components such as stabilization or "upwinding". A rectification step could then be used to go from the reduced solution to the 'truth' original solution. The underlying idea is that the two stabilization methods, even if they arrive to different solutions, still represent the same underlying solution.

- Another direction is to assume that the calibration process makes the complicated local components manageable by ROM. For instance, to ensure the TVD property, Finite Volume schemes often involve some gradient limitation in the vicinity of shocks. Can this be handled by standard ROM? Denote by $d_N(\nabla_{lim})$ it's Kolmogorov N-width in some norm. Suppose that the shocks are first order polynomials, whose coefficients vary in $A_0 := [a_0^{min}, a_0^{max}]$, $A_1 := [a_1^{min}, a_1^{max}]$. That is,
$$\forall\mu,\ \exists(a_0(\mu), a_1(\mu)) \in A_0 \times A_1,\ \text{s.t},\ \text{s}(y;\mu) = a_0(\mu) + a_1(\mu) * y.$$
Let h some characteristic size of the mesh. Then, we can estimate $d_N(\nabla_{lim})$ as
$$d_N(\nabla_{lim}) \approx \frac{mes(A_0) * mes(A_1)}{h^2}.$$
There is no hope in trying to handle this term using the Empirical Interpolation Method (EIM) but as calibration reduces the geometric variability of the shock position, the coefficients in the calibrated problem $\hat{A}_0$ and $\hat{A}_1$ will both be of order h. This drastically diminish the N-width of $d_N(\nabla_{lim})$. The same kind of arguments can be used for upwinding type of terms.

The first method will not be further discussed here, as the numerous mesh interpolations imply very high computational costs, as well as numerical errors. The third method is out of the scope of this chapter. We have chosen to prove the feasibility of the second method. It assumes the existence of a fully functioning CFD code. In the lines of what has been done [36], the idea is to keep the stability and accuracy properties of the existing code. The computational savings would be obtained using EIM/hyper reduction ideas.

The objective is to recast the original problem defined on Ω, onto an equivalent problem defined on $\hat{\Omega}$. This is a well studied problem in the elliptic and parabolic communities, see for instance [93,114]. It relies on the variational form of the PDE at hand. A similar procedure for our hyperbolic problem could be performed on a non conservative formulation. There are two issues with this approach in our setting. The first one is that this derivation is not rigorous as some of the quantities appearing are not properly defined for discontinuous solutions. Also, this formulation is not suited for our purpose, as the resulting problem is no longer posed as a conservation law, and thus require some intrusion into the CFD code. The intent here is to find a mapping procedure fitted for conservation laws. We will see that it involves a modifications of both flux and boundary conditions.

We start with a step common to Finite Volume schemes and Residual Distribution schemes. Let $\{\omega_i,\ i \in [1, \ldots, \mathcal{N}]\}$ the set of control volumes in Ω and let u any state variable. The integration of the conservation law in space and time, in the control volume i gives:

$$\int_{\omega_i} u(w, t^{n+1})dw - \int_{w_i} u(w, t^n)dw + \int_{\omega_i}\int_{t^n}^{t^{n+1}} \nabla \cdot \mathbf{f}(u)dtdw = 0. \tag{4.18}$$

Using arguments that are detailed in the appendix, we can show that equation (4.18) is equivalent to:

$$\int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^{n+1})|J_{F_n^{-1}}|d\hat{w} - \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^n)|J_{F_n^{-1}}|d\hat{w} + \int_{\hat{\omega}_i}\int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T\mathbf{f}(\hat{u}))dtd\hat{w} = 0 \tag{4.19}$$

where $\hat{u} := u \circ F_n^{-1}$, $\hat{\omega}_i = F_n(\omega_i)$ and $N_n^T\mathbf{f}$ is the correct modified flux with

$$N_n^T = \begin{bmatrix} (J_{F_n^{-1}})_{22} & -(J_{F_n^{-1}})_{12} \\ -(J_{F_n^{-1}})_{21} & (J_{F_n^{-1}})_{11} \end{bmatrix}_n,$$

where J_F denotes the Jacobian of any mapping F. This equality is known as the Piola transform, which is usually used in a different context, see for instance [91]. We will make the assumption that the determinant of the Jacobian is sufficiently smooth and the mesh is fine enough so that we can consider N_n^T constant per element. The error due to this approximation will not be investigated in here.

Remark 4.4.1. *Some more rigorous approaches could be developed, but would lead to more intrusion into the CFD code. In [42] for instance, they choose to work with the average of $\hat{u}|J_{F_n^{-1}}|$ over control volumes, instead of $\hat{u}$.*

We arrive to the following equation in each control volume $\hat{w}_i$.

$$\int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^{n+1})d\hat{w} - \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^n)d\hat{w} + \frac{1}{|J_{F_n^{-1}}|_i}\int_{\hat{\omega}_i}\int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T\mathbf{f}(\hat{u}))dtd\hat{w} = 0.$$

We have all the ingredients to feed the CFD code:

- a mesh: here it is the reference mesh, over $\hat{\Omega}$;
- the average of the solution over control volumes:

$$\hat{\mathbf{u}}_i = \frac{1}{\text{mes}(\hat{w}_i)}\int_{\hat{w}_i} \hat{u}(\hat{w}, t^n)$$

- a flux, in a closed form: with the Piola transform, here it just amounts to

$$N_n^T\mathbf{f}$$

  where the N_n^T term will depend on the time step and is not constant over $\hat{\Omega}$. We will see in Section 4.6.3 that using Gordon-Hall mapping type allows a proper offline/online decomposition

- boundary conditions: we do not need to worry about the outside boundary conditions, as they will not be affected by the mapping. The no slip boundary conditions for the original problem are simply obtained by setting

$$u \cdot \vec{\mathbf{n}} = 0 \text{ on the wing.}$$

In our case, these are imposed as follows: treat the boundary nodes as any other node, and substract the correct quantity to impose the no slip boundary condition. More precisely, let $\mathbf{n} = (n_1, n_2)$ the norm at the boundary . The flux at nodes on the boundary are given by:

$$(\mathbf{f}_x, \mathbf{f}_y) \cdot \mathbf{n} = \begin{pmatrix} \rho\Big((u,v)\cdot\mathbf{n}\Big) \\ \rho u\Big((u,v)\cdot\mathbf{n}\Big) + pn_1 \\ \rho v\Big((u,v)\cdot\mathbf{n}\Big) + pn_2 \\ \Big((u,v)\cdot\mathbf{n}\Big)(E+p) \end{pmatrix}$$

We enforce the no slip boundary condition by subtracting the following quantity:

$$(\tilde{\mathbf{f}}_x, \tilde{\mathbf{f}}_y) \cdot \mathbf{n} = \begin{pmatrix} \rho\Big((u,v)\cdot\mathbf{n}\Big) \\ \rho u\Big((u,v)\cdot\mathbf{n}\Big) \\ \rho v\Big((u,v)\cdot\mathbf{n}\Big) \\ \Big((u,v)\cdot\mathbf{n}\Big)(E+p) \end{pmatrix}$$

We can use the Piola transform again for these terms. The subtracted quantity formulated in terms of the reference variables is simply given by

$$\int_{\partial\hat{w}_i} (\tilde{\mathbf{f}}_x(\hat{u}), \tilde{\mathbf{f}}_y(\hat{u})) \cdot (N_n^T \cdot \mathbf{n}) d\hat{w}.$$

The conclusion from this analysis is that under the assumption that the determinant of the Jacobian is constant per element, changing the normals in the CFD code is enough to compute the total residual in each triangle. This is the first part of the RD scheme, see Section 4.2.3. What follows is the distribution of the residual among nodes, in each element. As mentioned in the offline section, the CFD code is of Lax-Friedriechs type. The residual is evenly distributed. This procedure is independent of the mesh and of the solution. There is no additional work. For an upwinding scheme, this is a much more difficult problem to tackle, being not in the scope of this work.

As mentioned in Section 4.2.3, the truth scheme uses SUPG type stabilization. We have not studied in this chapter how to modify this term in order to have an equivalent stabilization procedure on $\hat{u}$. We will discuss this approximation in the numerical experiment section.

We now assume that we have performed the $(n+1)$-th iteration with the CFD code. The output is denoted $\tilde{u}^{n+1}$ and by construction, $\tilde{u}^{n+1} \circ F_n \approx u^{n+1}$. As F_n is not, a priori, the right mapping for u^{n+1}, we are looking simultaneously for:

- a better suited mapping F_{n+1}

- the corresponding $\hat{u}^{n+1}$ expressed in terms of the reduced basis defined on $\hat{\Omega}$.

Following the lines of [34], define the following objective function, for $p \in \{1, 2\}$:

$$J^p : \begin{cases} \mathcal{F} \times \mathbb{R}^{N^{red}} & \rightarrow \quad \mathbb{R} \\ F, \{\alpha_k\}_k & \mapsto \quad \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^p(\Omega)} . \end{cases} \tag{4.20}$$

4.5 Finding the coordinates, for a fixed mapping

In this section, $\tilde{u}^{n+1}$ and F_n are considered fixed. We first propose an optimization procedure when the mapping F in (4.20) is assumed to be known. Fix $F \in \mathcal{F}$ and define J^p_F as:

$$J^p_F : \begin{cases} \mathbb{R}^{N^{red}} & \rightarrow \quad \mathbb{R} \\ \{\alpha_k\}_k & \mapsto \quad \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^p(\Omega)} \end{cases} \tag{4.21}$$

We are going to discuss two particular cases. First, the $p = 2$ case, which is the standard technique in ROM and then an extension to $p = 1$ minimization, which was advised in [5].

4.5.1 L^2-norm minimization, standard Galerkin projection

The objective functional is thus given by

$$J^2_F : \{\alpha_k, k \in [1, \ldots, N^{red}]\} \rightarrow \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^2(\Omega)}$$

and the first order optimality condition gives us the αs:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N^{red}} \end{pmatrix} = A \begin{pmatrix} < \tilde{u}^{n+1} \circ F_n, \phi_1 \circ F >_{L^2(\Omega)} \\ < \tilde{u}^{n+1} \circ F_n, \phi_2 \circ F >_{L^2(\Omega)} \\ \vdots \\ < \tilde{u}^{n+1} \circ F_n, \phi_{N^{red}} \circ F >_{L^2(\Omega)} \end{pmatrix},$$

where $A_{i,j} :=< \phi_i \circ F, \phi_j \circ F >_X$ is a symmetric invertible square matrix of size N^{red}. We have

$$\begin{aligned} \forall i, \ < \tilde{u}^{n+1} \circ F_n, \phi_i \circ F >_{L^2(\Omega)} \quad &= \quad \int_{\hat{\Omega}} \tilde{u}^{n+1} \phi_i \circ \delta_F |J_{F_n^{-1}}| \\ \forall i, j, \ < \phi_i \circ F, \phi_j \circ F >_{L^2(\Omega)} \quad &= \quad \int_{\hat{\Omega}} \phi_i \phi_j |J_{F^{-1}}|, \end{aligned}$$

where $\delta_F := F \circ F_n^{-1}$.

Remark 4.5.1. *One needs to take into account the fact that the basis $\{\phi_k \circ F\}_k$ will most probably not be an orthogonal basis.*

So far, we have replaced the expensive problem involving the absolute mapping by a problem where mappings are close to the identity. We will see in Section 4.6.3 how to achieve efficient offline/online decomposition.

4.5.2 L^1-norm minimization

The objective functional is here given by:

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J^1_F(\alpha) = \sum_{i=1}^{\mathcal{N}} \int_{\hat{\omega}_i} \left| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right|.$$

Once again, a standard change of variable gives:

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J^1_F(\alpha) = \sum_{i=1}^{\mathcal{N}} \int_{\hat{\omega}_i} \left| \tilde{u}^{n+1} \circ \delta_F^{-1} - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \right| |J_{F^{-1}}| \tag{4.22}$$

In each control volume i, we choose a set of N_i^{quad} quadrature points $\{\hat{x}_{i,j},\ j \in [1, \ldots, N_i^{quad}]\}$ and the corresponding weights $\{\gamma_{i,j},\ j \in [1, \ldots, N_i^{quad}]\}$. We have:

$$J^1_F(\alpha) = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{N_i^{quad}} \gamma_{i,j} \left| \tilde{u}^{n+1}(\delta_F^{-1}(\hat{x}_{i,j})) - \sum_{k=1}^{N^{red}} \alpha_k \phi_k(\hat{x}_{i,j}) \right| |J_{F^{-1}}(\hat{x}_{i,j})| . \tag{4.23}$$

This is handled as in [5] by recasting it as a linear programming problem. For now, the size of the problem is of order $\mathcal{N}$, the number of control volumes of the mesh. We will see in Section 4.6.3 how to reduce the computational cost.

4.6 Finding the mapping

One important remark is that the shock's position evolves smoothly in time. This is rigorously justified by Rankine-Hugoniot conditions. Let $\hat{A}_0$ and $\hat{A}_1$ be the maximum absolute values for the variation between two successive pseudo time steps of respectively position and slope of the shock. These are roughly given by:

$$\forall i \in \{0, 1\},\ \hat{A}_i \approx W^{1,\infty} \text{ (maximum shock speed).}$$

We use these values to define the following neighborhood of the identity in $\mathcal{F}$:

$$\mathcal{F}^{rel} := \left\{ GH(\hat{a}_0, \hat{a}_1),\ |\hat{a}_i| \leq \hat{A}_i \right\},$$

where the application GH has been defined in equation (4.17).

4.6.1 Alternative differentiable objective function

Let $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$. It is clear that for solutions with shocks, the following application is not smooth:

$$\begin{array}{ccc} \mathcal{F}^{rel} & \to & X \\ \delta_F & \mapsto & \hat{u}(\delta_F(\cdot)). \end{array}$$

More precisely, the derivative in the sense of distributions has a Dirac mass at the shock. We give here a formal proof, and we refer to [11, 19] for a rigorous one. Denote with

$\Sigma_0 := \Big\{(\hat{x},\hat{y}) \in \hat{\Omega},\ \text{ s.t } \hat{u} \text{ is discontinuous } \Big\}$ and $v \to [v]$ the standard jump operator.

By construction, Σ_0 is independent of $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$. Each solution in the calibrated solution manifold can be decomposed into a smooth component and one discontinuity:

$$\forall \hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}},\ \exists \hat{u}_{smooth} \text{ and } \hat{u}_j,\ \text{ s.t } \hat{u} = \hat{u}_{smooth} + [\hat{u}_j]|_{\Sigma_0}$$

$$\hat{u} - \hat{u} \circ \delta_F = \hat{u}_{smooth} - \hat{u}_{smooth} \circ \delta_F + [\hat{u}_j]|_{\Sigma_0} - [\hat{u}_j \circ \delta F]|_{\delta F^{-1}(\Sigma_0)}$$

The derivative in the sense of distributions has thus also two components:

$$\hat{u} - \hat{u} \circ \delta_F \approx \partial \hat{u}_{smooth} + \delta|_{\Sigma_0} \partial \Sigma$$

where $\delta|_{\Sigma_0}$ is the Dirac mass at Σ_0.

We propose one option to circumvent this issue, an alternative and differentiable objective function. For $\tilde{u}^{n+1}$ the output of one iteration of the CFD code, denote

$$\Sigma(\tilde{u}^{n+1}) := \Big\{(\hat{x},\hat{y}) \in \hat{\Omega} \text{ s.t, } \tilde{u}^{n+1} \text{ is discontinuous } \Big\}$$

As already mentioned, because of R-H condition, $\Sigma(\tilde{u}^{n+1})$ will be close to Σ_0. We use the $p = 1$ notation, but the following approach can be directly transposed to the $p = 2$ case. It is easy to see why for $\hat{x}_{i,j}$ sufficiently far from the shock so that

$$\forall \delta_F \in \mathcal{F}^{rel},\ \delta_F(\hat{x}_{i,j}) \text{ is on the same side of } \Sigma_0 \text{ as } \hat{x}_{i,j}, \tag{4.24}$$

the following application is differentiable:

$$\begin{cases} \mathcal{F}^{rel} & \to & \mathbb{R} \\ \delta_F & \mapsto & \tilde{u}^{n+1}(\delta_F(\hat{x}_{i,j})). \end{cases}$$

Following this remark, we denote by $\hat{\Omega}_d$ the subdomain of $\hat{\Omega}$, where we have removed some neighborhood of the shock. More precisely, let

$$\hat{\Omega}_d := \bigcup \hat{\omega}_i, \text{ for } i \in [1,\dots,\mathcal{N}],\ \text{ s.t } \forall j \in [1,\dots,N_i^{quad}],\ \hat{x}_{i,j} \text{ satisfies condition (4.24) }.$$

We denote by Ω_d it's counterpart in the physical domain.

Remark 4.6.1. *For the L^1 norm, the overall problem as presented is not differentiable. This can be solved using Huber type minimization instead of the raw L^1 [5].*

With this new notation, we replace the original problem J_F^p with the differentiable objective function $J^p_{\Omega_d,F}$. We can now perform standard optimization algorithm to get the desired mapping δ_F, as

$$\begin{array}{lcl} [-\hat{A}_0, \hat{A}_0] \times [-\hat{A}_1, \hat{A}_1] \times \mathbb{R}^{N^{red}} & \to & \mathbb{R} \\ \hat{a}_0, \hat{a}_1, \{\alpha_k\} & \mapsto & J_{\Omega_d,\text{G-H}(\hat{a}_0,\hat{a}_1)}(\alpha) \end{array} \tag{4.25}$$

is a smooth application.

Following the previous discussion, we define smaller objective functions i.e, for every subdomain Ω_{sub} of Ω_d, define the following functional $J_{\Omega_{sub},F}$:

$$J^p_{\Omega_{sub},F} : \{\alpha_k\}_k \to \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^p(\Omega_{sub})}.$$

4.6.2 One possible algorithm

We now present one way of performing in practice the optimization of the quantity defined in (4.25):

- discretize the set $[-\hat{A}_0, \hat{A}_0]$ and $[-\hat{A}_1, \hat{A}_1]$: $\{\hat{a}_0^k,\ k \in [1,\ldots,N_0]\}$ and $\{\hat{a}_1^k,\ k \in [1,\ldots,N_1]\}$.
- denote $\Psi_{\mathcal{F}^{rel}}$ the following sample of $\mathcal{F}^{rel}$:

$$\Psi_{\mathcal{F}^{rel}} := \{GH(\hat{a}_0^k,\ \hat{a}_1^p),\ k \in [1,\ldots,N_0],\ p \in [1,\ldots,N_1]\}.$$

- compute the coordinates for all mappings in $\Psi_{\mathcal{F}^{rel}}$ using the algorithm in Section 4.5 and deduce the corresponding values of the objective function:

$$\forall \delta_F \in \Psi_{\mathcal{F}^{rel}}, \quad \text{compute} \inf_{\alpha \in \mathbb{R}^{N^{red}}} J_{\Omega_d, \delta_F}(\alpha).$$

- interpolate the previously computed quantities to get an estimate of

$$\inf_{\alpha \in \mathbb{R}^{N^{red}}} J_{\Omega_d, \delta_F}(\alpha) \text{ over } \mathcal{F}^{rel}.$$

  Deduce the value of the optimal coefficients $\hat{a}_0^{opt}$ and $\hat{a}_1^{opt}$, as in [34].
- deduce the reduced coordinates for the corresponding mapping $GH(\hat{a}_0^{opt},\ \hat{a}_1^{opt})$ using the techniques in Section 4.5.

Remark 4.6.2. *Other ideas to find F_{n+1} can be implemented. They are however less natural in our framework.*

- ***Shock fitting:*** *close to what has been described in the offline section. Find the control volumes such that $\tilde{u}^{n+1}$ has highest gradient and fit a polynomial. This is made computationally efficient because we do not need to look for highest gradient all over Ω: $\Sigma(\tilde{u}^{n+1})$ is close to Σ_0.*
- ***RK condition:*** *update the shock positions and slopes using the explicit form of the shock speed given by Rankine-Hugoniot.*

4.6.3 Online/offline decomposition

We have not yet discussed the computational complexity of our full algorithm. For now, at each time step, we need to run the full CFD code to get $\tilde{u}^{n+1}$ over $\hat{\Omega}$. Until we manage to build a self sufficient reduced scheme, see the third method described in section 4.4 , this computational time is not easily reducible. The only ideas available in the literature are hyper reduction [123].

In the previous section, we have restricted the problem from Ω to Ω_d because of differentiability. Here, we replace Ω_d by an even smaller, denoted generically Ω_{sub} because of computational cost. Of course, we will look for Ω_{sub} subsets of Ω_d to keep the differentiability property. The hyper-reduction method is an empirical procedure that aims at selecting a "good" Ω_{sub}.

We present here a version of the hyper-reduction procedure that uses a different objective function than $J^p_{\Omega_{sub},F}$ defined in the previous section. Note that many different variants of this algorithm that we propose here are possible. For $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$, define

$$I^p_{\hat{\Omega}_{sub}}(\hat{u}) : \left\{\alpha_k,\ k \in [1, \ldots, N^{red}]\right\} \mapsto \left\| \hat{u} - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \right\|_{L^p(\hat{\Omega}_{sub})}.$$

During the hyper-reduction procedure, we try to find $\hat{\Omega}_{sub}$ such that:

$$\begin{aligned} \forall \hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}},\ & \underset{\{\alpha_k\} \in \mathbb{R}^{N^{red}}}{\operatorname{arginf}} I^p_{\hat{\Omega}_{sub}}(\hat{u}) \left(\left\{\alpha_k,\ k \in [1, \ldots, N^{red}]\right\}\right) \\ & \approx \underset{\{\alpha_k\} \in \mathbb{R}^{N^{red}}}{\operatorname{arginf}} I^p_{\hat{\Omega}_d}(\hat{u}) \left(\left\{\alpha_k,\ k \in [1, \ldots, N^{red}]\right\}\right). \end{aligned} \tag{4.26}$$

That is, we want the optimization not to be affected too much by the reduction of the size of the problem. Of course, we do not know the continuous set $\mathcal{M}_{\mathcal{F},\mathcal{D}}$. Let us denote $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$ a representative set of the continuous manifold, and let ϵ some threshold. We perform the following greedy algorithm.

Algorithm 4 One possible algorithm to select $\hat{\Omega}_{sub}$

Data: $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$,$\{\phi_k,\ k \in [1, \ldots, N^{red}]\}$
Result: $\hat{\Omega}_{hyper}$,N^{hyper}
Initialize $\hat{\Omega}_{hyper} := \bigcup_{i \in I_{ini}} \hat{\omega}_i$ **repeat**
  $\forall \hat{u} \in \Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}},\ \{\beta_k(\hat{u}), k \in [1, \ldots, N^{red}]\} := \underset{\left\{\alpha_k,\ k \in [1,\ldots,N^{red}]\right\}}{\operatorname{arginf}} I^p_{\hat{\Omega}_{hyper}}(\hat{u}) \left(\left\{\alpha_k,\ k \in [1, \ldots, N^{red}]\right\}\right)$;
  $i := \underset{p \in \mathcal{N}}{\operatorname{argsup}} \sup_{\hat{u} \in \Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}} \| \sum_{k=1}^{N^{red}} \beta_k(\hat{u}) \phi_k - \hat{u} \|_{L^p(\hat{\omega}_p)}$;
  $\hat{\Omega}_{hyper} := \hat{\Omega}_{hyper} \bigcup \hat{\omega}_i$;
until *convergence*;
$N^{hyper} := \text{card}(\hat{\Omega}_{hyper})$

The idea of hyper reduction is that on the solution manifold there is a one to one correspondence between the restriction of the solution on $\hat{\Omega}^{hyper}$ and the full solution. For the problem at hand, we expect that the solutions in the solution manifold are characterized by their behavior in the vicinity of the shock. Because of calibration, the knowledge of the solutions in a reduced number of control volumes around Σ_0, independent of μ, should thus be enough to completely characterize the solution. The accuracy of the overall process can be estimated but not guaranteed. Indeed, the greedy algorithm is performed on sampled spaces: $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$ instead of $\mathcal{M}_{\mathcal{F},\mathcal{D}}$ and $\Psi_{\mathcal{F}^{rel}}$ instead of $\mathcal{F}^{rel}$. One possible output of the algorithm is illustrated in Figure 4.9, where $\hat{\Omega}_{hyper}$ is the reunion of black elements. What the calibration has achieved, at the same time as it has reduced the N-width of the solution set, is to localize spatially the interesting part of the solutions. We can thus anticipate that the control volumes chosen will be accumulated around the calibrated shock, as depicted in Figure 4.10. We expect that calibration reduces the number of control volumes required to achieve some

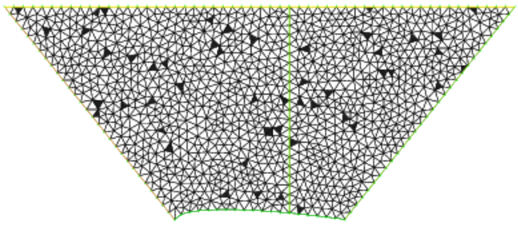


Figure 4.9: One possible output of Algorithm 4

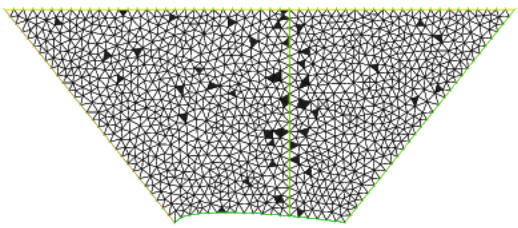


Figure 4.10: A more realistic output of Algorithm 4

prescribed accuracy.

For our choice of online implementation, the computation of the N^T terms is not an issue, as these are only required in a moderate number of cells, denoted by N^{hyper}. We will nevertheless emphasize that these terms, because of the choice of Gordon-Hall mapping, would not be a computational problem even with no hyper reduction. By inspecting the structure of the G-H mapping, see equation (4.16), we can see that the weights and the projection functions are not parameter dependent. Also, in our problem, $\mu \to \psi_i(\cdot;\mu)$ for $i \in \{1, 2, 3\}$

are linear function of μ. Most of the terms appearing in equation (4.16) are thus trivially affinely decomposable. The fact that the computation of terms involving ψ_4 also fall into the offline/online decomposition paradigm requires more work. We do not enter the details, but one could show it by using a variation of the G-H method (see section 4.7.2) and the fact that away from Γ_1, the wing can be approximated by a polynomial.

4.7 Numerical Experiments

In this chapter we focus on the numerical results of the following novelty: the resolution of an equivalent calibrated problem on a reference mesh using the Piola transform (see Section 4.4). Of course, the overall performance of such an approach relies on the ability to construct a smooth family of mappings $\mathcal{F}$. This has been challenging and is a big part of the numerical experiments presented below.

4.7.1 Mapping on a flat domain

The first experiment we discuss is a preliminary, alpha test: we try to reproduce one snapshot, using the Piola transform and a reference mesh. We are running the CFD code for Mach = 0.81 and AoA = 3.0°. The truth solution that we are trying to recover is presented in Figure 4.11. We first perform a 'control sample' test. We run the original CFD code on the

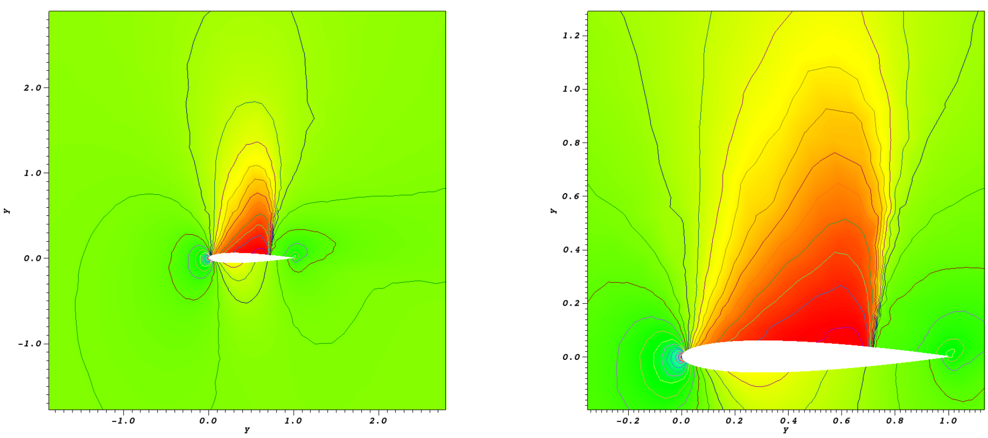


Figure 4.11: Truth solution for velocity component with Mach=0.81 and AoA=3.0°

reference mesh presented in Section 4.3. The output solution is presented in Figure 4.12. As expected, it is not comparable with the truth solution: $u(\mu)$. Indeed, the problem solved is not equivalent to the original one. We need to modify fluxes and boundary conditions, as presented in Section 4.4. As the steady solution $u(\cdot;\mu)$ is known, we can compute its shock position and slope: $a_0(\mu)$ and $a_1(\mu)$. We use the G-H mapping (4.16) on both $\hat{\Omega}_R$ and $\hat{\Omega}_L$ and then, run the modified CFD code. Note that with this preliminary approach, the N^T term

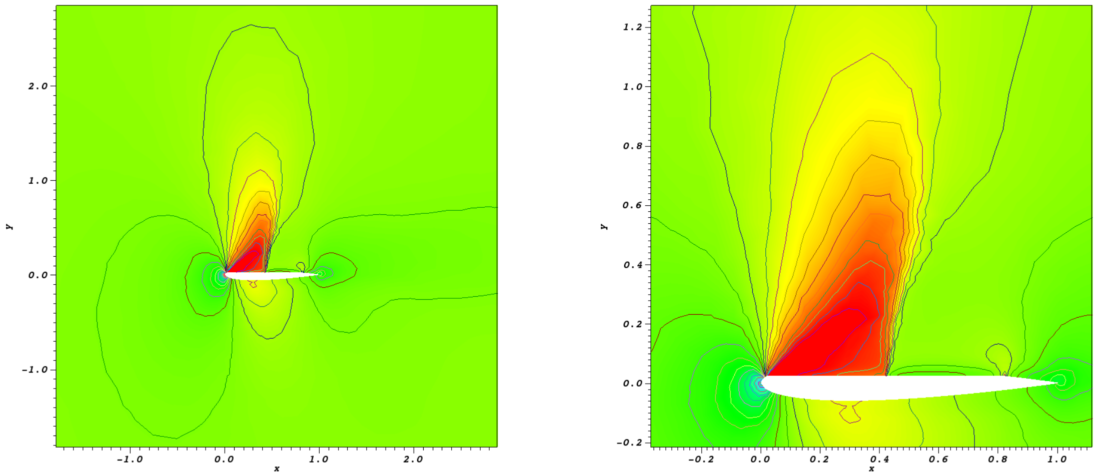


Figure 4.12: The identity mapping velocity component on a flat domain

is not updated at each pseudo time step. Figure 4.13 shows the resulting solution, that we denote $\hat{u}(\mu)$. One can observe that the general behavior is correct. The shock is more or less located at the correct position and it has been straightened. In other words, quantitatively we have $u(\mu) \circ GH(a_0(\mu), a_1(\mu)) \approx \hat{u}(\mu)$ on $\hat{\Omega}$. This preliminary result is a first answer to the viability of using the Piola transform to construct equivalent problems on reference meshes. Nevertheless, we can see that we have some non physical behavior close to the wing. This could have been anticipated, as the mapping constructed in Section 4.3 suffers major flaws. The biggest problem seems to be at the wing and a consequence of the high gradient at the bottom left corner of the left domain $\hat{\Omega}_L$. We conclude that we need a smoother mapping,

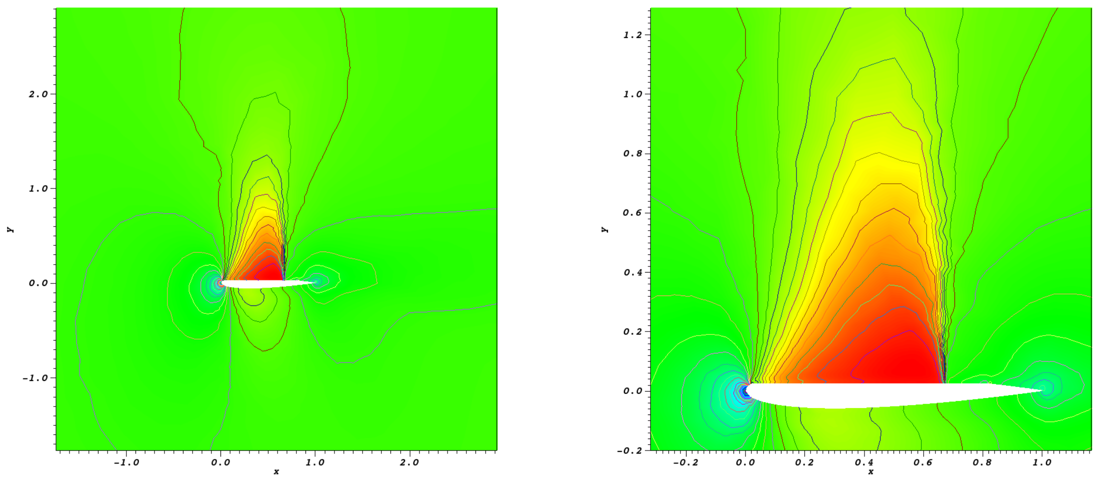


Figure 4.13: The mapped solution for velocity component on a flat domain

more than a continuous one on $\partial\hat{\Omega}_L$ and $\partial\hat{\Omega}_R$.

4.7.2 Mapping on a curved domain

The choice of a flat wing of the previous sections was intentional in order to remind that the reference domain on which we are solving the problem is not the physical one. Nevertheless, because of the lack of smoothness of the resulting mapping, we have decided to use a more advanced mapping than the raw Gordon-Hall. Our starting point is the method developed in [92], applied to the domains depicted on Figures 4.14 and 4.15. As in Section 4.3, to enforce continuity of the global mapping, we require that the four corners of reference and physical domain match, i.e we require:

$$\begin{aligned}(x_1, y_1) &= (\hat{x}_1, \hat{y}_1)\\ (x_5, y_1) &= (\hat{x}_5, \hat{y}_1)\\ (x_4, y_2) &= (\hat{x}_4, \hat{y}_2)\\ (x_2, y_2) &= (\hat{x}_2, \hat{y}_2)\end{aligned}$$

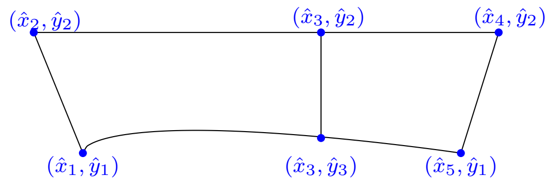


Figure 4.14: Reference domain $\hat{\Omega}$

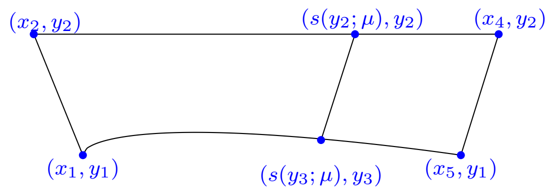


Figure 4.15: Physical domain Ω

4.7.2.1 Original formulation

This extension of the G-H mapping, also called generalized transfinite extension in the literature, has the same structure as the original G-H. For each boundary on the reference domain, $\hat{\Gamma}_i$, we need one parametrization of the physical counterpart Γ_i, that is

$$\psi_i \circ \pi_i|_{\hat{\Gamma}_i} : \begin{cases} \hat{\Gamma}_i & \rightarrow \quad \Gamma_i \\ (\hat{x}, \hat{y}) & \mapsto \quad (x, y). \end{cases}$$

The mapping is then taken as a weighted combination of these mapped boundaries :

$$GH(\hat{x},\hat{y}) = \sum_{i=1}^{4}[\phi_i(\hat{x},\hat{y})\psi_i(\pi_i(\hat{x},\hat{y}),\mu) - \phi_i(\hat{x},\hat{y})\phi_{i+1}(\hat{x},\hat{y})\psi_i(1,\mu)]. \tag{4.27}$$

where ϕ_i and π_i are respectively the weight and the projection functions associated to $\hat{\Gamma}_i$ (see section 4.3.2). The linear weights and the projection functions are not an option anymore, as the reference domain $\hat{\Omega}$ is not a rectangle.

We will first present the choice of the weights and the projections proposed in the original version [92]. This was done in a very general case, and the focus was put on the smoothness of the overall mapping. The weights functions are taken as the solutions of the following Laplace problems:

$$\forall i \in [1,\dots,4], \begin{cases} -\Delta\phi_i & = & 0 & \text{in } \hat{\Omega} \\ \phi_i & = & 1 & \text{on } \hat{\Gamma}_i \\ \phi_i & = & 0 & \text{on } \hat{\Gamma}_{i+2} \\ \frac{\partial\phi_i}{\partial n} & = & 0 & \text{on } \hat{\Gamma}_{i-1}\cup\hat{\Gamma}_{i+1}. \end{cases} \tag{4.28}$$

The projection functions are also chosen as solutions to a Laplace problem :

$$\forall i \in [1,\dots,4], \begin{cases} -\Delta\pi_i & = & 0 & \text{in } \hat{\Omega} \\ \pi_i & = & t & \text{on } \hat{\Gamma}_i, t \text{ monotone and smooth} \\ \pi_i & = & 1 & \text{on } \hat{\Gamma}_{i+1} \\ \pi_i & = & 0 & \text{on } \hat{\Gamma}_{i-1} \\ \frac{\partial\pi_i}{\partial n} & = & 0 & \text{on } \hat{\Gamma}_{i+2}. \end{cases} \tag{4.29}$$

Remark 4.3.1 on the bijectivity of the resulting mapping still holds in this extended version of the Gordon-Hall method.

4.7.2.2 Additional ingredients

We will now deal with the issues mentioned in Section 4.3 one by one. The smoothness of $\partial\hat{\Omega}$ is solved, with the new choice of $\hat{\Omega}$. Also, we had noticed in our flawed flat approximation that missing to take into account the curvature of the wing represents a too rough approximation. We thus need to chose π_4 and ψ_4 accordingly. The proper way of dealing with this curved boundary is to use the standard arclength definition. For instance, the projection function on the wing π_4 is chosen as :

$$\pi_4|_{\Gamma_4} : (\hat{x},\hat{y}) \rightarrow \int_0^{\hat{x}} \sqrt{1+\left(\frac{\partial w}{\partial \hat{x}}\right)^2},$$

the same holding for ψ_4.

We also need to be closer to the identity mapping on the left boundary. In the original formulation, homogeneous Neumann boundary conditions are imposed on neighboring edges when computing the projection functions defined in (4.29). This choice is not the right one for our particular problem. We present in Figure 4.16 on the left, the projection function

π_3 in the transfinite version of [92]. Remember, π_3 is the projection onto the edge $\hat{\Gamma}_3$. It is clear that this particular choice deforms the coordinate system. This is one of the causes of the lack of smoothness of the mapping on the left edge $\hat{\Gamma}_1$. The right picture in Figure 4.16 presents π_3 for a better suited boundary condition.

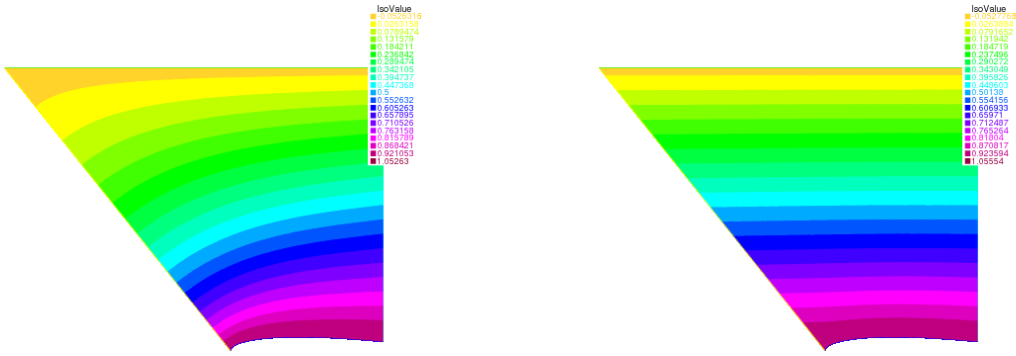


Figure 4.16: Left: π_3 in the original formulation, with homogeneous Neumann boundary condition; Right: π_3 for a more suitable boundary condition

Towards the same objective, we do not want any stretching of the solution around the left boundary and close to the shock. Indeed, we need smooth transitions to neighboring domains. In order to enforce this, one necessary step is to modify ψ_2 and ψ_4 from the original version. Denote with $H(x)$ some smoothed Heaviside step function. We write it for ψ_2, but the same can be done for ψ_4. We pick the following :

$$\tilde{\psi_2}(\hat{x},\hat{y},\mu) = \pi_2(\hat{x},\hat{y})\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\cdot(1-H(\pi_2(\hat{x},\hat{y})))+\left(1+(\pi_2(\hat{x},\hat{y})-1)\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\right)\cdot H(\pi_2(\hat{x},\hat{y})).$$

That is, we want no stretching for $\pi_2(\hat{x},\hat{y}) \approx 0$ or 1. A graphical illustration is presented on the left picture of Figure 4.17 for an hypothetical stretching of 4/3. The dashed red lines correspond to a non stretched mapping.

We will modify one more ingredient. We take steeper weight functions for boundaries 1 and 3. For instance, we can pick

$$\tilde{\phi_1}(\hat{x},\hat{y}) = H\left(\phi_1(\hat{x},\hat{y})\right),$$

where the ϕ_1 is the solution the the Laplace problem (4.28). This is presented on the right picture of Figure 4.17. What this achieves is that close to left boundary, the exact shape of the right physical boundary has no influence, and the converse.

Finally, we choose the following set of $\{\psi_i,\ i\in[1,\ldots,4]\}$:

$$\begin{aligned}
\psi_1(\pi_1(\hat{x},\hat{y}),\mu) &:= \Big(x_1+\pi_1(\hat{x},\hat{y})\cdot(x_2-x_1), y_1+\pi_1(\hat{x},\hat{y})\cdot(y_2-y_1)\Big)\\
\psi_2(\pi_2(\hat{x},\hat{y}),\mu) &:= \Big(x_2+\pi_2(\hat{x},\hat{y})\cdot(s(y_2;\mu)-x_2), y_2\Big)\\
\psi_3(\pi_3(\hat{x},\hat{y}),\mu) &:= \Big(s(y_2+\pi_3(\hat{x},\hat{y})\cdot(y_3-y_2);\mu), y_2+\pi_3(\hat{x},\hat{y})\cdot(y_3-y_2)\Big)\\
\psi_4(\pi_4(\hat{x},\hat{y}),\mu) &:= \Big(arclen^{-1}(\pi_4(\hat{x},\hat{y})), y_3\Big)
\end{aligned}$$

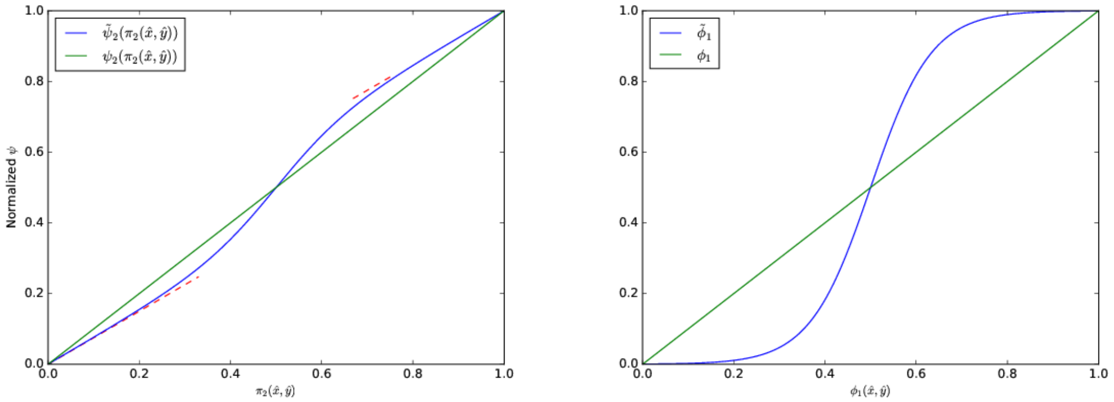


Figure 4.17: Modification of weights and projection functions to get smoother transitions on $\hat{\Gamma}_1$ and $\hat{\Gamma}_3$

Remark 4.7.1. *The offline/online decomposition of the global method will strongly depend on the way we pick the set of ϕ_i's, π_i's and ψ_i's.*

Remark 4.7.2. *This smarter choice of functions not only makes the G-H mapping smoother but it also makes*

$$\begin{array}{ccc} \mathcal{D} & \to & \mathcal{F} \\ \mu & \mapsto & F_\mu \end{array}$$

smoother. This can be an interesting property in a optimal control context.

To assess the gain of this more advanced mapping, we perform the same test as in Subsection 4.7.1, this time with the new and improved G-H mapping, given by equation (4.27). The output solution $\hat{u}(\mu)$ is presented in Figure 4.18. As for the results obtained with the original G-H, the overall behavior is correct as $\hat{u}(\mu)$ has a similar shape as $u(\mu) \circ GH(a_0(\mu), a_1(\mu))$. The novelty is that we have managed to remove the non physical behavior at the boundary that we had in the raw G-H scenario (see Figure 4.13).

Remark 4.7.3. *One must not forget that this case is no different from the flat boundary scenario of subsection 4.7.1. The fact that the reference domain has the same body as the physical domain is required for smoothness purposes only.*

Before a more involved test run, we present yet another improvement. This goes one step further in building a smooth mapping at the boundaries. One recent development on transfinite maps is defined in [74] and is called boundary displacement dependent transfinite map (BDD TM). The idea is not to construct the whole mapping, but to construct a relative displacement with respect to the identity. Most of the method is the same, the only difference is that instead of ψ_i function, which represent the position on the physical domain, a new function $d_i : [0,1] \times \mathcal{D} \to \mathbb{R}$ is introduced and it will represent the displacement:

$$d_i(t,\mu) = \psi_i(t,\mu) - \hat{\psi}_i(t).$$

Each of the boundaries in the reference domain is parametrized by $\hat{\psi}_i : [0,1] \to \mathbb{R}$. Like this, the mapping will take into account the original positions of the points in the reference domain

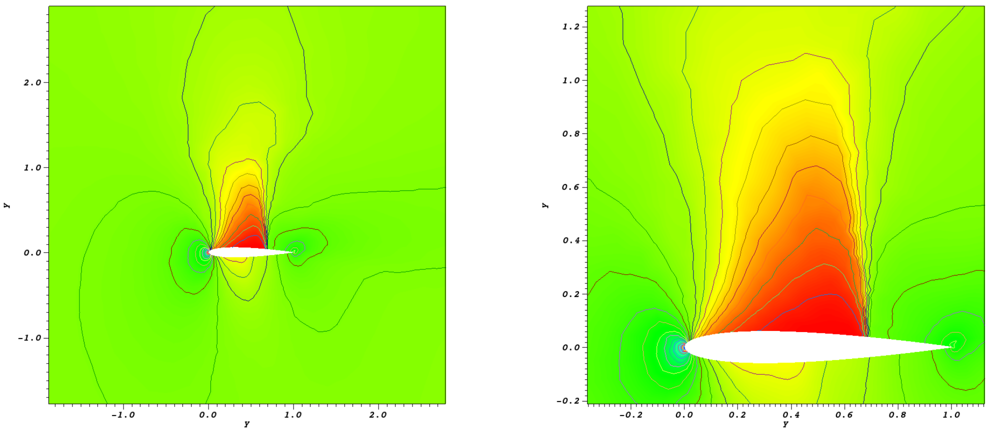


Figure 4.18: The mapped solution for velocity component on a curved domain

$\hat{\Omega}$ and will move them by weighting only the difference between the original boundaries and the deformed ones. Let $(\hat{x}, \hat{y})$ a point in the reference domain $\hat{\Omega}$, the idea of BDD TM is to displace it through the quantity $(\hat{x}, \hat{y}) + \sum_{i=1}^{n} \phi_i(\hat{x}, \hat{y}) d_i(\pi_i(\hat{x}, \hat{y}), \mu)$. In the end, the BDD transfinite mapping is defined as:

$$GH_{BDDTM}(\hat{x}, \hat{y}) = (\hat{x}, \hat{y}) + \sum_{i=1}^{n} \Big(\phi_i(\hat{x}, \hat{y}) d_i(\pi_i(\hat{x}, \hat{y}), \mu) - \phi_i(\hat{x}, \hat{y}) \phi_{i+1}(\hat{x}, \hat{y}) d_i(1, \mu) \Big) \quad (4.30)$$

This has one major effect, on the left boundary for instance, where we want zero displacement. The resulting mapping restricted to a neighborhood of this boundary will be the identity, which guarantees overall smoothness.

Remark 4.7.4. *The improvements on ϕ's and ψ's presented for the TM method still apply to the BDD TM.*

After this long preamble, we are ready to illustrate numerically the gain obtained from the methods just described. We present in Figure 4.19 the comparison between the original method described in Section 4.7.2.1 and the one tailored for our specific application. We show one of the entries of N^T. We have picked the most varying one i.e $\frac{\partial x}{\partial \hat{x}}$. It is obvious that the previous improvements to G-H have helped for the smoothness between neighboring domains.

4.7.3 Final experiment

What is presented in this section does not correspond to any actual step of the online section. The purpose is to provide a more quantitative result on the utilization of the Piola transform for resolution of a problem on a reference mesh. For this, we have chosen to perform the following test:

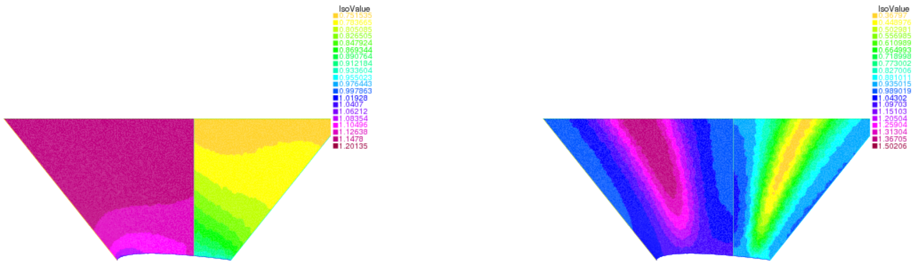


Figure 4.19: One of the entries of the Jacobian matrix, namely $(J_{F_n^{-1}})_{11}$. Left: with no additional smoothing ingredients; Right: with some smoothing ingredients

- pick a small number of pairs $\{(a_0, a_1)\}$ and construct the corresponding G-H mappings: $\{GH(a_0, a_1)\}$;
- as in the previous subsection, launch the CFD code, using the modified flux and boundary conditions. Denote the output $\hat{u}(a_0, a_1)$ for each mapping $GH(a_0, a_1)$. Once again, the mapping is not updated at each pseudo time step;
- compare the output with the mapped 'truth' solution, i.e compare

$$u \circ (GH(a_0, a_1)) \text{ with } \hat{u}(a_0, a_1).$$

We have chosen a simple comparison criteria: the position and the slope of the shock. We present the results for two pairs (a_0, a_1) in Figure 4.20. Blue represents the shock of the truth solution mapped onto the reference domain, $u \circ GH(a_0, a_1)$, red is the shock of $\hat{u}(a_0, a_1)$ and green is the position of the shock of u, which has been plotted for control purposes. We have fitted one degree polynomials through each shock. The discrepancy on the left picture,

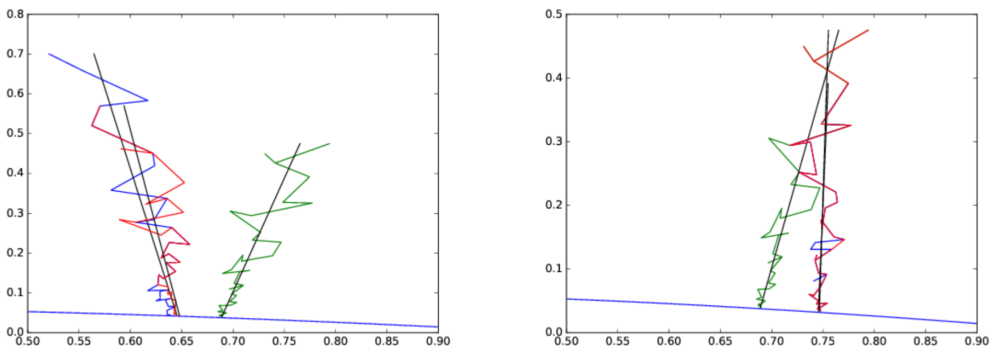


Figure 4.20: Comparison of the outputs

between the output of the modified CFD code and the mapped truth scheme can be due to many factors:

- numerical errors on the computation of the N^T terms;
- the SUPG stabilization has not been touched, to avoid too much intrusion in the code. This means that we are not using the same stabilization procedure as the truth scheme. We refer to [96] for a study of this situation. They advise using an a posteriori procedure, called rectification;
- our method to locate the shock is basic. We would need something more involved to quantify the error.

Appendix

The objective in this appendix is to prove the equivalence of equations (4.18) and (4.19). We reformulate the problem in a general setting. Let two domains Ω and $\hat{\Omega}$, $F_n : \Omega \to \hat{\Omega}$ some smooth mapping and

$$\hat{w} = F_n(w) \Rightarrow w = F_n^{-1}(\hat{w}).$$

We have some generic smooth flux defined on Ω, denoted by $\mathbf{f} : \Omega \to \mathbb{R}^2$. The objective is to prove that there exists a vector field $\tilde{\mathbf{f}} : \hat{\Omega} \to \mathbb{R}^2$, which is a function depending on $\mathbf{f}$ and of F_n such that

$$\nabla_w \cdot \mathbf{f} = \frac{1}{J_{F_n^{-1}} \circ F_n} \left(\nabla_{\hat{w}} \cdot \tilde{\mathbf{f}}(\hat{w}) \right) \circ F_n \text{ on } \Omega, \tag{4.31}$$

where as usual, J_F denotes the Jacobian of the mapping F. Here

$$J_{F_n^{-1}}(\hat{w}) = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix}.$$

We will first compute the representation of $\mathbf{f}$ in terms of curvilinear coordinates and the frame

$$\boldsymbol{\rho}_1(\hat{w}) = \left(\frac{\partial x}{\partial \hat{x}}, \frac{\partial y}{\partial \hat{x}} \right), \boldsymbol{\rho}_2(\hat{w}) = \left(\frac{\partial x}{\partial \hat{y}}, \frac{\partial y}{\partial \hat{y}} \right)$$

associated with them. Thus, we treat the vector field $\mathbf{f}(w)$ by first expressing it in the form: $\mathbf{f} \circ F_n^{-1}(\hat{w}) = \hat{f}_1(\hat{w})\boldsymbol{\rho}_1 + \hat{f}_2(\hat{w})\boldsymbol{\rho}_2$. We denote with (f_1, f_2) the two components of $\mathbf{f}$. We have:

$$\begin{pmatrix} f_1 \circ F_n^{-1} \\ f_2 \circ F_n^{-1} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{pmatrix} \begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \end{pmatrix}$$

for some $\hat{f}_1$ and $\hat{f}_2$. Solving the system of equations, we obtain the following expressions for $\hat{f}_1$ and $\hat{f}_2$:

$$\begin{cases} \hat{f}_1 : \hat{w} & \mapsto & \frac{1}{\frac{\partial x}{\partial \hat{x}}\frac{\partial y}{\partial \hat{y}} - \frac{\partial x}{\partial \hat{y}}\frac{\partial y}{\partial \hat{x}}} \left(\frac{\partial y}{\partial \hat{y}} f_1 \circ F_n^{-1}(\hat{w}) - \frac{\partial x}{\partial \hat{y}} f_2 \circ F_n^{-1}(\hat{w}) \right) \\ \hat{f}_2 : \hat{w} & \mapsto & \frac{1}{\frac{\partial x}{\partial \hat{x}}\frac{\partial y}{\partial \hat{y}} - \frac{\partial x}{\partial \hat{y}}\frac{\partial y}{\partial \hat{x}}} \left(-\frac{\partial y}{\partial \hat{x}} f_1 \circ F_n^{-1}(\hat{w}) + \frac{\partial x}{\partial \hat{x}} f_2 \circ F_n^{-1}(\hat{w}) \right). \end{cases}$$

Then, we have

$$\nabla_w \cdot \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \nabla_w \cdot (\hat{f}_1 \circ F_n \ \boldsymbol{\rho}_1 \circ F_n) + \nabla_w \cdot (\hat{f}_2 \circ F_n \ \boldsymbol{\rho}_2 \circ F_n).$$

Using the product rule, we obtain for $i \in \{1,2\}$:

$$\begin{aligned}\nabla_w \cdot (\hat{f}_i \circ F_n\ \boldsymbol{\rho}_i \circ F_n) &= \nabla_w \left(J_{F_n^{-1}} \circ F_n\ \hat{f}_i \circ F_n\right) \cdot \left(\tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right) \\ &+ \left(J_{F_n^{-1}} \circ F_n\ \hat{f}_i \circ F_n\right) \nabla_w \cdot \left(\tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right).\end{aligned}$$

The second operand is zero. We will sketch the proof and refer to [42] for a rigorous one. We consider ϕ any smooth real valued test functions with compact support in Ω. We want to show the following :

$$\forall \phi \in \mathcal{D}(\Omega),\ \int_\Omega \phi(w) \nabla_w \cdot \left(\frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right) dw = 0.$$

For this, we use the product rule :

$$\begin{aligned}\forall \phi \in \mathcal{D}(\Omega),\ \textstyle\int_\Omega \phi(w) \nabla_w \cdot \left(\tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right) dw &= \textstyle\int_\Omega \nabla_w \cdot \left(\phi(w) \tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right) dw \\ &- \textstyle\int_\Omega \tfrac{1}{J_{F_n^{-1}} \circ F_n} \left(\boldsymbol{\rho}_i \circ F_n\right) \cdot \nabla_w \phi(w) dw.\end{aligned}$$

With the first integral one the right hand side is easy to deal by just using the divergence theorem and the fact that ϕ has a compact support in Ω:

$$\int_\Omega \nabla_w \cdot \left(\phi(w) \frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n\right) dw = 0.$$

The other equality requires more attention:

$$\begin{aligned}\textstyle\int_\Omega \tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n(w) \cdot \nabla_w \phi(w) dw &= \textstyle\int_{\hat{\Omega}} \tfrac{1}{J_{F_n^{-1}}} \boldsymbol{\rho}_i \cdot \nabla_w \left(\phi \circ F_n^{-1}\right)(\hat{w}) |J_{F_n^{-1}}| d\hat{w} \\ &= \pm \textstyle\int_{\hat{\Omega}} \boldsymbol{\rho}_i \cdot \nabla_w \left(\phi \circ F_n^{-1}\right)(\hat{w}) d\hat{w}.\end{aligned}$$

We will use the following equality two times :

$$\forall \psi \in \mathcal{D}(\hat{\Omega}),\ \nabla_w \psi \cdot \boldsymbol{\rho}_1 = \frac{\partial \psi}{\partial x}\frac{\partial x}{\partial \hat{x}} + \frac{\partial \psi}{\partial y}\frac{\partial y}{\partial \hat{x}} = \frac{\partial \psi}{\partial \hat{x}}. \tag{4.32}$$

This is precisely the reason why we have defined $\{(\boldsymbol{\rho}_0, \boldsymbol{\rho}_1)\}$ in the first place. The previous equality becomes then:

$$\int_\Omega \frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_1 \circ F_n(w) \cdot \nabla_w \phi(w) dw = \int_{\hat{\Omega}} \frac{\partial(\phi \circ F_n^{-1})}{\partial \hat{x}}(\hat{w}) d\hat{w},$$

by using the fact that $\phi \circ F_n^{-1}$ has compact support in $\hat{\Omega}$.

We finally have the announced result :

$$\forall i \in \{1,2\},\ \nabla_w \cdot (\hat{f}_i \circ F_n\ \boldsymbol{\rho}_i \circ F_n) = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_w \left(J_{F_n^{-1}} \circ F_n \hat{f}_i \circ F_n\right) \cdot (\boldsymbol{\rho}_i \circ F_n)$$

Using the same argument as in (4.32), we have

$$\sum_{i=1}^{2} \nabla_w \cdot (\hat{f}_i \circ F_n\ \boldsymbol{\rho}_i \circ F_n) = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_{\hat{w}} \left(J_{F_n^{-1}} \hat{f}_i\right) \circ F_n$$

We conclude that

$$\nabla_w \cdot \mathbf{f} = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_{\hat{w}} \cdot \left(N_n^T \mathbf{f}\right) \circ F_n.$$

Hence, we have found the desired flux function of equation (4.31), $\tilde{\mathbf{f}} := N_n^T \cdot \mathbf{f}$, where

$$N_n^T = \begin{bmatrix} (J_{F_n^{-1}})_{22} & -(J_{F_n^{-1}})_{12} \\ -(J_{F_n^{-1}})_{21} & (J_{F_n^{-1}})_{11} \end{bmatrix}.$$

In the end, we obtain the desired equivalence:

$$(4.18) \Leftrightarrow \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^{n+1}) |J_{F_n^{-1}}(\hat{w})| d\hat{w} - \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^n) |J_{F_n^{-1}}(\hat{w})| d\hat{w} + \int_{\hat{\omega}_i} \int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T \mathbf{f}(\hat{w})) dt d\hat{w} = 0.$$

Chapter 5: Reduction of the computational cost with applications in UQ

5.1 Introduction

Parametrized partial differential equations (PPDE) have received in the last decades an increasing amount of attention from research fields as engineering and applied sciences. All these domains have in common the dependency of the PPDE on the input parameters, which are used to describe possible variations in the solution, initial conditions, source terms and boundary conditions, to name just a few. Hence, the solutions of these problems are depending on a large number of different input values, as in optimization, control, design, uncertainty quantification, real time query and other applications. In all these cases, the aim is to be able to evaluate in an accurate and efficient way an output of interest when the input parameters are varying. This will be very time consuming or can even become prohibitive when using high-fidelity approximation techniques, such as finite element (FE), finite volume (FV) or spectral methods. For this kind of problems, model order reduction (MOR) techniques are used, in order to replace the high-fidelity problem by one featuring a much lower numerical complexity. A key ingredient of MOR are the reduced basis (RB) methods, which allow to produce fast reduced surrogates of the original problem by only combining a few high-fidelity solutions (*snapshots*) computed for a small set of parameter values [61, 73, 108]. The most common and efficient strategies available to build a reduced basis space are the proper orthogonal decomposition (POD) and the greedy algorithm. These two sampling techniques have the same objective but in very different approach forms: the POD method allows to compress large snapshot sets to the most important POD modes, that means a few vectors or functions containing the most important information of the data [31,32,76,82,115,130]. This POD-based MOR technique was successfully applied to 4D parameter spaces for full-blown, viscous, CFD computations pertaining to an aircraft CRM [137]. On the other side, greedy algorithm [111,112,122] is based on an iterative sampling from the parameter space, fulfilling at each step a suitable optimality criterion that relies on a posteriori error estimates.

A first challenge in the context of ROM deal with unsteady problems, so implicitly the exploration of a parameter-time framework is needed. In this case, the sampling strategy to construct reduced basis spaces for the time-dependent problem is POD-greedy [66] and is based on combining the POD algorithm in time, with a greedy algorithm in the parameter space. Thus, we search the currently worst resolved parameter using an error bound or indicator, then compute the complete trajectory of the corresponding solution, orthogonalize this trajectory to the current RB-space, perform a POD with respect to time in order to compress the error trajectory to its most important new information, and add the new POD-mode to the current basis.

A second challenge refers to the nonlinear problems. The simple "polynomial" nonlinearities, can be written as a multilinear form in the variational form of the PDE, where this multilinearity can be effectively used for suitable offline/online decomposition of the Galerkin-reduced system and the Newton-type iteration for solving the fixpoint equation. Also, a-posteriori error analysis is possible for these RB-approaches making use of the Brezzi-Rappaz-Raviart theory. Problems that can be treated by this are nonlinear diffusion or nonlinear advection problems, e.g. the Burgers Equation. For more general nonlinearities, the Empirical Interpolation Method (EIM) has been introduced as a general function interpolation technique and has been applied to stationary and instationary nonlinear problems. This method was first introduced in [21] and in the context of ROM in [61]. Some applications of the EIM method are discussed in [97] and an a posteriori error analysis is presented in [51, 61]. There are only a few papers in the literature which are focused on MOR methods for parametric nonlinear hyperbolic conservation laws and they are based on: POD and Galerkin projection [77, 118], domain partitioning [128], Gauss-Newton with approximated tensors (GNAT) [37], L^1-norm minimization [5] or suitable algorithms extended to linear and nonlinear hyperbolic problems [66, 67]. The work of Drohmann, Haasdonk and Ohlberger [50], presents a new approach of treating nonlinear operators in the reduced basis approximations of parametrized evolution equations based on empirical interpolation namely, the PODEI-Greedy algorithm, which constructs the reduced basis spaces for the empirical interpolation in a synchronized way.

In this chapter, we focus on reduced order models for hyperbolic conservation laws based on explicit finite volume (FV) schemes. The FV schemes will be formulated within the framework of residual distribution (RD) schemes. The advantages of this alternative are: a better accuracy, a much more compact stencil, easy parallelization, explicit scheme and no need of a sparse mass matrix "inversion". For more details on RD, we refer to the work of Abgrall [2–4]. However, we want to emphasize that our approach can be applied to any general FV formulation and RD is just our choice. In this work, we concentrate on uncertainty quantification (UQ) applications for hyperbolic conservation laws. In practice, the input parameters are obtained by measurements (observations) and these measurements are not always very precise, involving some degree of uncertainty [26, 54]. A good example of hyperbolic conservation laws is when computing the flow past an airfoil or a wing, the inputs for this calculation, such as the inflow Mach number, the angle of attack, as well as the parameters that specify the airfoil geometry, are all measured with some uncertainty. This uncertainty in the inputs results in the propagation of uncertainty in the solution [9]. Moreover, the need of model order reduction for UQ is obvious by just taking into account that these problems feature high-dimensionality, low regularity and arbitrary probability measures. However, the classical methods (Monte Carlo, stochastic Galerkin projection method, stochastic collocation method, etc) can not be applied directly to solve the underlying deterministic PDEs, since they might need millions of full solutions (or even more), in order to achieve a certain accuracy. Hence, with the help of reduced basis method, together with an a posteriori error estimate, we will be able to break the curse of dimensionality of solving high dimensional UQ problems whenever the quantities of interest reside in a low dimensional space. Up to our knowledge, there is no work done on MOR for hyperbolic conservation laws with applications in UQ and the only results that are available in literature are holding for elliptic PDEs [39–41].

In the first section we will present the problem of interest namely, the unsteady hyperbolic conservation laws and we will explain the RD scheme in relation with the nonlinear fluxes. In Section 5.3 we will describe the algorithms that we are using for the construction of the reduced basis: POD-Greedy, PODEI. In Section 5.4 we describe the UQ method and in the last Section we present our numerical results.

5.2 Problem of interest

5.2.1 Hyperbolic conservation laws

In this work, we consider high-dimensional models (HDM) arising from the space discretization of hyperbolic PPDEs. These problems are characterized by a parameter $\boldsymbol{\mu} \in \mathcal{P}$ from some set of possible parameters $\mathcal{P} \subset \mathbb{R}^p$. The unsteady problem then consists of determining the state variable solution $\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})$ on a bounded interval $D \subset \mathbb{R}^d, d = 1, 2, 3$ and finite time interval $\mathbb{R}_+ = [0, T], T > 0$ such that the system of $m, m \geq 1$ balance laws to be satisfied of type (2.53):

$$\begin{cases} \mathbf{u}_t(\mathbf{x}, t; \boldsymbol{\mu}) + \mathcal{L}(\mathbf{x}, t; \boldsymbol{\mu})[\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})] & = \mathbf{h}(t; \boldsymbol{\mu}),\ \mathbf{x} \in D,\ t \in \mathbb{R}_+, \\ \mathbf{B}(\mathbf{u}; \boldsymbol{\mu}) & = \mathbf{g}(t; \boldsymbol{\mu}),\ \mathbf{x} \in \partial D,\ t \in \mathbb{R}_+, \\ \mathbf{u}(\mathbf{x}, t = 0; \boldsymbol{\mu}) & = \mathbf{u}_0(\mathbf{x}; \boldsymbol{\mu}),\ \mathbf{x} \in D, \end{cases} \tag{5.1}$$

where the operator $\mathcal{L}(\cdot, t; \boldsymbol{\mu}) = \mathrm{div} f(\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}))$ represents the divergence of the nonlinear flux $\mathbf{f} : \mathbb{R}^m \to (\mathbb{R}^m)^d$, $\mathbf{B}$ is a suitable boundary operator, and $\mathbf{h}, \mathbf{g}$ are volume, respectively surface forces. Obviously, the moving shocks and discontinuities will depend on the different parameter settings $\boldsymbol{\mu} \in \mathcal{P}$ and will develop during time. The task of the RB method will be to capture the evolution of both smooth and discontinuous solutions.

The discrete evolution schemes are based on approximating high-dimensional discrete space $\mathcal{W}_h \subset L^2(D)$ (or subset of some Hilbert space), $\dim(\mathcal{W}_h) = N_h$, where h represents the characteristic mesh size and by approximating the exact solution at time-instances $0 = t^0 < t^1 < \dots t^K = T$ i.e providing a sequence of functions $\mathbf{u}_h^k(\boldsymbol{\mu}) : \mathbb{R}^{N_h} \to \mathbb{R}^m$ for $k = 0, \dots, K$ such that $\mathbf{u}_h^k(\boldsymbol{\mu}) \approx \mathbf{u}(t_k; \boldsymbol{\mu})$.

5.2.2 Residual distribution scheme

In this section, we are interested in the class of RD methods and we will show how any FV scheme can be written in this framework. We consider D_h to be the triangulation of the domain D (see Figure 5.1), $\Delta t_k = t_{k+1} - t_k$ the time steps for $k = 0, \dots, K$ and we denote by T a generic element of the mesh. We define the set $\sum_h := \{\boldsymbol{\tau}_i\}_{i=1}^{N_h} \subset \mathcal{W}_h^{'}$ of linearly independent functionals, which are unisolvent on $\mathcal{W}_h$ i.e, there exist unique functions $\rho_i \in \mathcal{W}_h,\ i = 1, \dots, N_h$ and satisfy:

$$\boldsymbol{\tau}_j(\rho_i) = \delta_{ij},\ 1 \leq j \leq N_h.$$

The linear functionals $\boldsymbol{\tau}_i, i = 1, \dots, N_h$ are called the degrees of freedom (DoFs) of the discrete function space $\mathcal{W}_h$, equipped with a scalar product $\langle \cdot, \cdot \rangle_{\mathcal{W}_h}$ and a norm $|| \cdot ||_{\mathcal{W}_h}$, and the functions $\rho_i, i = 1, \dots, N_h$ are called the basis or shape functions. This shape functions can be for e.g, finite element, finite volume or discontinuous Galerkin basis functions on a numerical grid $D_h \subset D$.

In this case, the solution approximation space $\mathcal{W}_h$ is given by globally continuous piecewise polynomials of degree r:

$$\mathcal{W}_h = \{\mathbf{u} \in L^2(D_h) \cap C^0(D_h), \mathbf{u}_{|T} \in \mathbb{P}^r, \forall\ T \in D_h\} \tag{5.2}$$

so that the numerical solution $\mathbf{u}_h^k$ can be written as a linear combination of shape functions $\rho_i \in \mathcal{W}_h,\ i = 1, \ldots, N_h$.

The main steps of the RD methods can be summarized as follows:

1. For any element $T \in D_h$, compute the total residual
$$\Phi^T = \int_T \text{div } (\mathbf{f}_h(\mathbf{u}_h))d\mathbf{x} = \int_{\partial T} \mathbf{f}_h(\mathbf{u}_h) \cdot \vec{\mathbf{n}}\ d\tilde{\mathbf{x}}, \tag{5.3}$$
where $\mathbf{f}_h$ is an approximation of $\mathbf{f}$ (Figure 5.2).
2. For any DoF $\boldsymbol{\tau}$ within an element T, define the nodal residuals $\Phi_{\boldsymbol{\tau}}^T$ as the contribution to the fluctuation term Φ^T (Figure 5.3) such that:
$$\sum_{\boldsymbol{\tau} \in T} \Phi_{\boldsymbol{\tau}}^T = \Phi^T. \tag{5.4}$$
Equivalently, denoting by $\beta_{\boldsymbol{\tau}}^T$ the distribution coefficient of the DoF $\boldsymbol{\tau}$, we obtain:
$$\beta_{\boldsymbol{\tau}}^T = \frac{\Phi_{\boldsymbol{\tau}}^T}{\Phi^T} \tag{5.5}$$
with
$$\sum_{\boldsymbol{\tau} \in T} \beta_{\boldsymbol{\tau}}^T = 1. \tag{5.6}$$
3. Assemble all the residual contributions $\Phi_{\boldsymbol{\tau}}^T$ from all elements T surrounding a node $\boldsymbol{\tau} \in D_h$ (Figure 5.4):
$$\sum_{T|\boldsymbol{\tau} \in T} \Phi_{\boldsymbol{\tau}}^T = 0,\ \forall \boldsymbol{\tau} \in D_h. \tag{5.7}$$

This is a very general formulation and many classical schemes can be formulated within this framework. This variability hides mostly in how the residual of each triangle is distributed among the DoFs $\boldsymbol{\tau} \in T$, that is, on the choice of $\beta_{\boldsymbol{\tau}}^T$. For instance, distributing it evenly among nodes corresponds to a Lax-Friedriech type of scheme and can be defined without any reference to the geometry of a control volume, only by using the physical structure of the local flow. Another example, is the finite volume schemes, which are constructed using directions that are only related to the mesh definition and not to the structure of the solution. In this case, and whatever the order of accuracy of the scheme is, the approximation $\mathbf{f}_h(\mathbf{u}_h)$ is defined as the Lagrange interpolant of $\mathbf{f}(\mathbf{u})$ at the DoF $\boldsymbol{\tau} \in T$.

5.3 Algorithm

Before starting discussing the full algorithm we have used for our method, we should point out which are the main difficulties that we will encounter preparing our reduced basis space RB.

First of all, we know that the main prerequisite of a RB method is the separability into an affine decomposition, where the parameter dependent functionals are evaluated separately

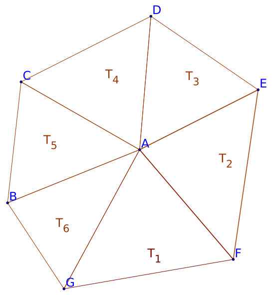


Figure 5.1: Triangulation D_h

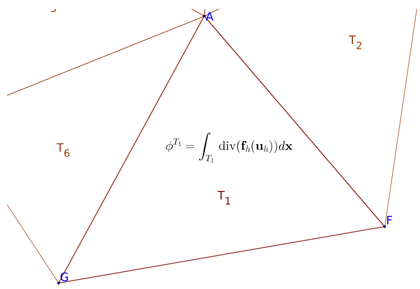


Figure 5.2: Compute the total residual

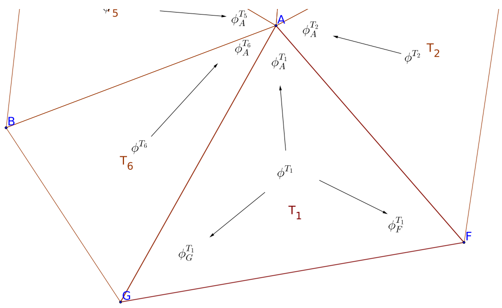


Figure 5.3: Compute the nodal residuals

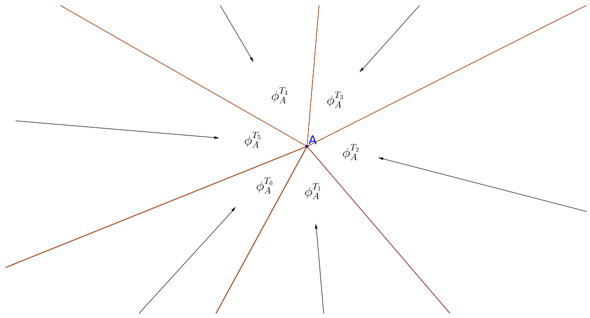


Figure 5.4: Collect all the residual contributions

with respect to some precomputed parameter independent operators. To efficiently apply this principle to a non–linear functional, like our $\mathcal{L}(\mathbf{x}, t; \boldsymbol{\mu})[\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})]$, we need to introduce the empirical interpolation method in order to approximate an (*a priori*) nonlinear parametrized operator with a separable one, which is efficient for evaluations of these operators for a reduced basis algorithm. We will show that this kind of surrogate operator can be computed in an efficient way using RD (or any FV) scheme in Section 5.3.5. Moreover, we need to build an efficient algorithm that will select sequentially some snapshots from some high–fidelity discretized solutions, until a prescribed tolerance. To do this, we will recur to a POD–Greedy algorithm, which is a combination of POD algorithm in time and a Greedy algorithm in the parameter space.

We will discuss in a general way the Greedy algorithm which was presented in Section 2.2.2.4, since also EIM and POD–Greedy can be recast into a Greedy algorithm.

5.3.1 Greedy algorithm

As seen in Section 2.2.2.4, a Greedy algorithm [111, 112] is taking as an input some given precomputed functions and is building a reduced basis space, where the error of the approximation of any of these snapshots into this reduced basis space is smaller than a certain prescribed tolerance. The way the algorithm is choosing the reduced basis space, is an iterative method. At each step, the Greedy algorithm is selecting the snapshot that is worst approximated by the reduced basis projection and it is enriching the reduced basis space adding this new function. There are 3 main procedures that we will use in the Greedy algorithm. They are:

- InitBasis which initializes the reduced basis $\mathcal{D}_N$, also called dictionary in literature;
- ErrorEstimate which estimates the error between the high–fidelity function and its projection on the reduced basis space $\mathcal{D}_N$;
- UpdateBasis which updates the RB space $\mathcal{D}_N$, given a certain selected parameter.

The greedy algorithm proceeds as in Algorithm 5.

Algorithm 5 Greedy Algorithm

Require: Training set $\mathcal{M}_{train} = \{\boldsymbol{\mu}_i\}_{i=1}^{N_{\text{train}}}$, tolerance ε^{tol} and N_{max}.
Ensure: Reduced basis $\mathcal{D}_N$
Initialize a reduced basis of dimension N_0:
  $\mathcal{D}_{N_0}$= InitBasis
  $N = N_0$
while $\max_{\boldsymbol{\mu}\in\mathcal{M}_{train}}$ErrorEstimate$(\mathbf{u}(\boldsymbol{\mu}), \mathcal{D}_N) \geq \varepsilon^{tol}$ AND $N \leq N_{max}$ **do**
  Find the parameter of worst approximated snapshot:
  $\boldsymbol{\mu}_{max} = \text{argmax}_{\boldsymbol{\mu}\in\mathcal{M}_{train}}$ ErrorEstimate$(\mathbf{u}(\boldsymbol{\mu}), \mathcal{D}_N)$
  Extend reduced basis $\mathcal{D}_N$ with the found snapshot (adding the new snapshot to dictionary):
  $\mathcal{D}_N, N =$UpdateBasis$(\mathcal{D}_N, \mathbf{u}(\boldsymbol{\mu}_{max}))$
end while

5.3.2 Empirical Interpolation Method

In this section we will apply the EIM algorithm [21] described in Section 2.2.2.5 to the discretized operators. The method has the goal to apply an interpolation to the fluxes $\mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})[\mathbf{u}(\mathbf{x}, t^k; \boldsymbol{\mu})] = \mathcal{L}^k(\mathbf{x}, t^k; \boldsymbol{\mu})[\mathbf{u}_h^k(\boldsymbol{\mu})]$. The set of the interpolant DoFs $\boldsymbol{\Sigma}_{N_{\text{EIM}}} = \{\boldsymbol{\tau}_m^{EIM}\}_{m=1}^{N_{\text{EIM}}}$, where $\boldsymbol{\tau}_m^{\text{EIM}} \in \mathcal{W}_h'$ and the corresponding set of interpolating basis functions $\mathcal{Q}_{N_{\text{EIM}}} = \{\mathbf{q}_m\}_{m=1}^{N_{\text{EIM}}}$, where $\mathbf{q}_m \in \mathcal{W}_h$ and $\boldsymbol{\tau}_m(\mathbf{q}_n) = \boldsymbol{\delta}_{mn}$ for $m \leq n$, will be the outputs of the algorithm. When the degrees of freedom can be identified with points in the domain (i.e. for Lagrange polynomial basis functions), EIM DoFs will be called “magic points”. The specialization of Greedy algorithm into the EIM algorithm consists in the definition of the greedy procedures, i.e. Algorithm 6, where the reduced basis, that we want to produce, comprise the interpolation DoFs $\boldsymbol{\Sigma}_{N_{\text{EIM}}}$ and the interpolation functions $\mathcal{Q}_{N_{\text{EIM}}}$, (i.e. $\mathcal{D}_N = (\mathcal{Q}_N, \boldsymbol{\Sigma}_N)$). After the EIM procedure, we will use the interpolated fluxes instead of the high fidelity discretized ones.

$$\mathcal{I}_{N_{\text{EIM}}}[\mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})][v_h] = \sum_{m=1}^{N_{\text{EIM}}} \boldsymbol{\tau}_m^{\text{EIM}} \left(\mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})[v_h]\right) \mathbf{q}_m \approx \mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})[v_h]. \tag{5.8}$$

The algorithm produced a basis $\mathcal{Q}_{N_{\text{EIM}}}$ which fulfills in a relaxed way the Kronecker's delta condition: $\boldsymbol{\tau}_m^{N_{\text{EIM}}}(\mathbf{q}_n) = \boldsymbol{\delta}_{mn}$ only if $m \leq n$. This condition will provide an upper triangular matrix that can be easily inverted during the EIM procedure to solve the interpolant coefficients problem. Moreover, the EIM basis functions spaces will be hierarchical, i.e. $\mathcal{Q}_M \subset \mathcal{Q}_{M+1}$, and the infinity norm of all the basis functions will be equal to 1 ($||\mathbf{q}_m||_\infty = 1$). Let us remark that, when we are dealing with Lagrange polynomial basis functions, formula (5.8) requires the evaluation of functions $\mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})[v_h]$ only in the *magic points*, and this will give the biggest reduction in computational time, since the evaluation of fluxes can be

Algorithm 6 Empirical Interpolation Method

EIM–InitBasis()

1: **return** empty initial basis $\mathcal{D}_0 = \emptyset$

EIM–ErrorEstimate$((\mathcal{Q}_M, \mathbf{\Sigma}_M), \boldsymbol{\mu}, t^k)$

1: Compute the exact flux $\mathbf{v}_h = \mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})][\mathbf{u}_h^k(\boldsymbol{\mu})]$

2: Compute the interpolation coefficients $\sigma^M(\mathbf{v}_h) := (\sigma_j^M)_{j=1}^M \in \mathbb{R}^M$ by solving the linear system (upper triangular)

$$\sum_{j=1}^{M} \sigma_j^M(\mathbf{v}_h)\boldsymbol{\tau}_i^{\mathrm{EIM}}[\mathbf{q}_j] = \boldsymbol{\tau}_i^{\mathrm{EIM}}[\mathbf{v}_h], \quad \forall i = 1, \ldots, M \tag{5.9}$$

3: **return** approximation error $||\mathbf{v}_h - \sum_{j=1}^M \sigma_j^M(\mathbf{v}_h)\mathbf{q}_j||_{\mathcal{W}_h}$

EIM–UpdateBasis $((\mathcal{Q}_M, \mathbf{\Sigma}_M), \boldsymbol{\mu}_{max}, t^{k_{max}})$

1: Compute the exact flux
$\mathbf{v}_h = \mathcal{L}(\mathbf{x}, t^{k_{max}}; \boldsymbol{\mu}_{max})][\mathbf{u}_h^{k_{max}}(\boldsymbol{\mu}_{max})]$

2: Compute the interpolation coefficients
$\sigma^M(\mathbf{v}_h) := (\sigma_j^M)_{j=1}^M \in \mathbb{R}^M$ from (5.9)

3: Compute the residual between the truth flux and its interpolant
$\mathbf{r}_M = \mathbf{v}_h - \sum_{j=1}^M \sigma_j^M(\mathbf{v}_h)\mathbf{q}_j$

4: Find the DoF that maximize the residual
$\boldsymbol{\tau}_{M+1}^{\mathrm{EIM}} := \mathrm{argmax}_{\boldsymbol{\tau}\in\mathbf{\Sigma}_h} |\boldsymbol{\tau}(\mathbf{r}_M)|$

5: Normalize the correspondent basis function
$\mathbf{q}_{M+1} := \boldsymbol{\tau}_{M+1}^{\mathrm{EIM}}(\mathbf{r}_M)^{-1} \cdot \mathbf{r}_M$

6: **return** updated basis $\mathcal{D}_{M+1} := ((\mathbf{q}_m)_{m=1}^{M+1}, (\boldsymbol{\tau}_m^{\mathrm{EIM}})_{m=1}^{M+1})$

very expensive. Indeed, the number of interpolation DoFs should be $N_{\mathrm{EIM}} \ll N_h$. In RD framework, we can explicitly see what we need to compute:

$$\boldsymbol{\tau}_i[\mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})][\mathbf{u}_h^k(\boldsymbol{\mu})] = \sum_{T|i\in T} \mathbf{\Phi}_i^T(\mathbf{u}_h^k(\boldsymbol{\mu})). \tag{5.10}$$

Each nodal residual $\mathbf{\Phi}_i^T(\mathbf{u}_h^k(\boldsymbol{\mu}))$ depends only on DoFs of element T, this means that for each *magic point* i we have to keep track of the function $\mathbf{u}_h^k(\boldsymbol{\mu})$ in all the DoFs of the elements T to which i belongs. The number of these DoF is *mesh–dependent*, for the simplest example in 1D with $\mathbb{P}^1$ piecewise continuous elements we know that for each magic point we have to keep track of 3 points: itself, its right and left neighborhoods. If we suppose some regularities on the mesh we can say that at most each vertex belongs to C elements. In this case, again for $\mathbb{P}^1$ Lagrangian basis functions, the number of DoF we are interested in is $R = C(K-2)+1$, where K is the biggest number of vertices that an element T can have.
At the end, we will have that the empirical interpolation method will provide an approximated version of the fluxes that depends at most on $RN_{\mathrm{EIM}} \ll N_h$ DoFs.

5.3.3 POD–Greedy

To create a reduced basis RB space, we want to find a low dimensionality *good* approximation of the high fidelity functional space $\mathcal{W}_h$. The algorithm that will provide this is a combination of different algorithms, such as POD [75, 82], POD–greedy [69], EIM–greedy [21]. What we will get is a POD–EIM–greedy algorithm, described by [50]. The main idea is to extend EIM basis functions and POD–greedy basis functions in a synchronized way, at each step of the main greedy algorithm.

A key ingredient of the procedure is the POD method, which is also known as PCA (principal component analysis) in statistical environment. The POD receives as input a set of vectors and returns the subspace of dimension N_{POD} which best represents the vectors given as a projection onto this subspace. We can write it in this way

$$POD(\{\mathbf{v}_i\}_{i=1}^N) = \underset{U|\dim(U)=N_{\text{POD}}}{\operatorname{argmin}} \max_{i\in\{1,\dots,N\}} \left(||\mathbf{v}_i - \mathcal{P}_U(\mathbf{v}_i)||_2\right). \tag{5.11}$$

Equivalently, this can be seen as the subspace of fixed dimension that maximizes the variance. The algorithm is based on SVD decomposition. We need to order the eigenvalues from the biggest to the smallest and we keep the first N_{POD} ones and the related eigenvectors. The span of the latter will be the output of the algorithm. To choose the dimension of this subspace, it is possible to use a tolerance, which will decide which percentage of the variance we want to keep or which percentage of the error we want to ignore. In our algorithms, we will use different tolerances, according to whether we want them to be fast (bigger N_{POD}) or sharp (small N_{POD}, even 1).

Before explaining the main algorithm, let us introduce the POD-Greedy algorithm, which deals with unsteady problems in the reduced basis context. The goal of the algorithm is to select new basis functions iteratively between precomputed snapshots $\{\{\mathbf{u}_h^k(\boldsymbol{\mu}_i)\}_{k=1}^K\}_{i=1}^{N_{\text{train}}}$. So, we have to find strategies to go through the parameter space and through the time steps. First, we explore the parameter space through a Greedy algorithm. We pick the parameter $\boldsymbol{\mu}_{max}$ that is worst approximated in RB space. Hence, on its temporal evolution $\{\mathbf{u}_h^k(\boldsymbol{\mu}_{max})\}_{k=1}^K$, we perform a POD that chooses the most representative M–dimensional space for that solution, to compress the solution in a few synthetic basis functions. Then we add to the RB space the new basis functions selected by POD. Finally, we perform a second POD on the RB space, to get rid of useless information.
Overall, we will compute a Greedy algorithm on the parameter domain $\mathcal{P}$ and a POD on the temporal space. Also in this case, we can write the POD-Greedy Algorithm 7, specifying the greedy procedures as in Algorithm 5.

Algorithm 7 POD–Greedy

POD–Greedy–InitBasis()

1: Pick a parameter $\boldsymbol{\mu}$ and compute the solution through all the time steps t^k: $\{\mathbf{u}_h^k(\boldsymbol{\mu})\}_{k=1}^K$
2: **return** initial basis $\mathcal{D}_0 = \text{POD}(\{\mathbf{u}_h^k(\boldsymbol{\mu})\}_{k=1}^K)$

POD–Greedy–ErrorEstimate(RB, $\boldsymbol{\mu}, t^k$)

1: **return** error indicator $\eta^k_{N,N_{\text{EIM}}}(\boldsymbol{\mu}) \geq ||\mathbf{u}_h^k(\boldsymbol{\mu}) - \mathbf{u}_N^k(\boldsymbol{\mu})||_{\mathcal{W}_h}$

POD–Greedy–UpdateBasis (RB, $\boldsymbol{\mu}_{max}$)

1: Compute the exact solution for all timestep with high fidelity solver $\{\mathbf{u}_h^k(\boldsymbol{\mu}_{max})\}_{k=1}^K$
2: Compute the Galerkin projection of the solution onto the RB space $\mathcal{P}[\mathbf{u}_h^k(\boldsymbol{\mu}_{max})]$
3: Compute the POD over time steps of the orthogonal projection of the high fidelity solution
$\text{RB}_{add} = \text{POD}\left(\{\mathcal{P}[\mathbf{u}_h^k(\boldsymbol{\mu}_{max})] - \mathbf{u}_h^k(\boldsymbol{\mu}_{max})\}_{k=1}^K\right)$
4: Compute a second POD to get rid of extra information
$\text{RB} = \text{POD}(\text{RB}_{add} \cup \text{RB})$
5: **return** updated basis RB

Let us point out a couple of details of Algorithm 7. At the beginning, we may initialize the reduced basis with a POD with a N_{POD} bigger than one used later (or a smaller error tolerance), since we still do not have any RB and we want to accelerate the first steps, to decrease the number of greedy steps. During the rest of the algorithm we will use the POD on the time evolution of the worst approximated solution in the training set and N_{POD} here will be smaller (or the tolerance will be bigger). The last POD that we use is in the last step of the POD–Greedy–UpdateBasis, where N_{POD} will be big and set by a very small tolerance (of the order of the final error that we want to reach). This will kill some spurious vectors that may come from oscillations or small errors. Often this step is not changing the updated reduced basis.

About the error indicator η, we would like to have a function which is independent of N_h that can be computed also in an *online phase*. Of course, this bound should also be enough sharp, to give a precise idea of the error. We will describe in section 5.3.6, an error indicator that is possible to use. If this indicator is not available, in the *offline phase* we can still use the real error, which is computationally less efficient, and in the *online phase*, where the high fidelity solutions are not available, we can not compute it directly. So, we will not have an explicit error bound to guarantee a good approximation.

In Algorithm 7, it is not written explicitly the EIM–method that every time we are applying to some reduced basis solutions. Moreover, the error indicator should also include the error produced by EIM procedure. This approach has some drawbacks described in [50]:

1. Is not really clear what is the relation between the tolerance used to stop EIM algorithm and the error produced in the POD–Greedy and how it influences the error indicator η. Therefore, it is impossible to determine a priori an optimal correlation between the reduced basis space and the EIM space.
2. The empirical interpolation error estimation depends on high dimensional computations for each parameter and time step tested. This can be very inefficient.

5.3.4 PODEIM–Greedy

To avoid these drawbacks, the idea of [50] is to synchronize the EIM and the POD–Greedy algorithms. We sketch the steps of the PODEIM-Greedy in Algorithm 8 with the remark that also this algorithm can be rewritten in terms of a greedy one 5.

The differences between this new algorithm and the POD–Greedy are in the update phase, where we enrich at the same moment the EIM and the RB basis. Moreover, it is possible that the error (and the indicator η) is not monotonically decreasing as the dimension of RB increases. This is caused by a bad approximation of the non–linear fluxes through the EIM. Indeed, in such a situation, we are enlarging only the EIM space and discarding the additional part of the RB that we added. This leads to an automatic tuning between N and N_{EIM}.

5.3.5 Online–phase

In this section we will describe the reduced basis scheme that we will eventually apply to find a reduced solution. This process is also used in the *offline–phase* at each greedy step for each

Algorithm 8 PODEIM–Greedy

PODEIM–Greedy–InitBasis()

$(\mathcal{Q}_{M_{small}}, \boldsymbol{\Sigma}_{M_{small}}) = \text{EIM-Greedy}(\mathcal{M}_{train}, \varepsilon_{tol,small})$
Pick a parameter $\boldsymbol{\mu}$ and compute the solution through all the time steps t^k: $\{\mathbf{u}_h^k(\boldsymbol{\mu})\}_{k=1}^K$
$\text{RB}_0 = \text{POD}(\{\mathbf{u}_h^k(\boldsymbol{\mu})\}_{k=1}^K)$
return initial bases $\mathcal{D}_0 = (\text{RB}_0, (\mathcal{Q}_{M_{small}}, \boldsymbol{\Sigma}_{M_{small}}))$

PODEIM–Greedy–ErrorEstimate($\mathcal{D}_S, \boldsymbol{\mu}, t^k$)

return error indicator $\eta^k_{N,N_{\text{EIM}}}(\boldsymbol{\mu})$

PODEIM–Greedy–UpdateBasis ($\mathcal{D}_S, \boldsymbol{\mu}_{max}$)

Extend EIM basis $D^{\text{EIM}}_{N_{\text{EIM}}+1} = \text{EIM–UpdateBasis } (D^{\text{EIM}}_{N_{\text{EIM}}}, \boldsymbol{\mu}_{max})$
Extend RB basis $D^{\text{RB}}_{N+1} = \text{POD–Greedy–UpdateBasis } (D^{\text{RB}}_{N}, \boldsymbol{\mu}_{max})$
Discard extended RB if error increases:
if $\eta^k_{N-1,N_{\text{EIM}}-1}(\boldsymbol{\mu}_{max}) < \max_{\boldsymbol{\mu}_i \in \mathcal{M}_{train}} \eta^k_{N,N_{\text{EIM}}}$ **then**
    return only EIM updated basis: $\mathcal{D}_{S+1} = (D^{\text{RB}}_N, D^{\text{EIM}}_{N_{\text{EIM}}+1})$
else
    return updated basis $\mathcal{D}_{S+1} = (D^{\text{RB}}_{N+1}, D^{\text{EIM}}_{N_{\text{EIM}}+1})$
end if

parameter in the training set, to get the reduced solution and the correspondent error. We will focus on explicit finite volume method, that can be rewritten into RD explicit scheme, but it is possible to extend this scheme to implicit (Newton iteration based method) as done in [50]. The basic idea is to replace the discrete evolution operator $\mathcal{L}[\cdot] := \mathcal{L}(\mathbf{x}, t^k; \boldsymbol{\mu})[\cdot]$ with its empirical interpolants and project it onto the RB space. For this purpose, let us introduce the orthogonal projection $\Pi : \mathcal{W}_h \to \text{RB}$ such that

$$\langle \Pi[u], \boldsymbol{\varphi} \rangle_{\mathcal{W}_h} = \langle u, \boldsymbol{\varphi} \rangle_{\mathcal{W}_h}, \qquad \forall \boldsymbol{\varphi} \in \text{RB} \tag{5.12}$$

and we can define the reduced operator as

$$\mathcal{L}_{\text{RB}} := \Pi \circ \mathcal{I}_{N_{\text{EIM}}} \circ \mathcal{L}. \tag{5.13}$$

Let us define $\{\boldsymbol{\varphi}_{\text{RB},i}\}_{i=1}^N$ a basis of RB, $\{\mathbf{q}_m\}_{m=1}^{N_{\text{EIM}}}$ the interpolation functions of EIM space and, for $m = 1, \dots, N_{\text{EIM}}$, let us define $\{\theta_i^m\}_{i=1}^N$ such that $\Pi(\mathbf{q}_m) = \sum_{i=1}^N \theta_i^m \boldsymbol{\varphi}_{\text{RB},i}$.

To begin the procedure, for any parameter $\boldsymbol{\mu}$, we compute the trajectory of the reduced solution, projecting the initial data onto the RB space: $\mathbf{u}_N^0(\boldsymbol{\mu}) := \Pi[\mathbf{u}_h^0(\boldsymbol{\mu})]$. Then, for each time step, we compute the reduced solution applying the reduced operator $\mathcal{L}_{\text{RB}}[\mathbf{u}_N^k]$. This implies to compute

$$
\begin{aligned}
\mathbf{u}_N^{k+1}(\boldsymbol{\mu}) =& \mathbf{u}_N^k(\boldsymbol{\mu}) - \mathcal{L}_{\text{RB}}[\mathbf{u}_N^k(\boldsymbol{\mu})] = \sum_{i=1}^N \alpha_{\text{RB},i}^k(\boldsymbol{\mu})\boldsymbol{\varphi}_{\text{RB},i} - \Pi(\mathcal{I}_{N_{\text{EIM}}}(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])) = \\
=& \sum_{i=1}^N \alpha_{\text{RB},i}^k(\boldsymbol{\mu})\boldsymbol{\varphi}_{\text{RB},i} - \Pi\left(\sum_{m=1}^{N_{\text{EIM}}} \boldsymbol{\tau}_m^{N_{\text{EIM}}}(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])\mathbf{q}_m\right) = \\
=& \sum_{i=1}^N \alpha_{\text{RB},i}^k(\boldsymbol{\mu})\boldsymbol{\varphi}_{\text{RB},i} - \sum_{m=1}^{N_{\text{EIM}}} \boldsymbol{\tau}_m^{N_{\text{EIM}}}(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])\Pi\left(\mathbf{q}_m\right) = \\
=& \sum_{i=1}^N \alpha_{\text{RB},i}^k(\boldsymbol{\mu})\boldsymbol{\varphi}_{\text{RB},i} - \sum_{m=1}^{N_{\text{EIM}}} \boldsymbol{\tau}_m^{N_{\text{EIM}}}(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])\sum_{i=1}^N \theta_i^m \boldsymbol{\varphi}_{\text{RB},i} = \\
=& \sum_{i=1}^N \left(\alpha_{\text{RB},i}^k(\boldsymbol{\mu}) - \sum_{m=1}^{N_{\text{EIM}}} \boldsymbol{\tau}_m^{N_{\text{EIM}}}(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])\theta_i^m\right)\boldsymbol{\varphi}_{\text{RB},i}.
\end{aligned}
\tag{5.14}
$$

In the last formula, what we really need to compute *online* is only $\boldsymbol{\tau}_m(\mathcal{L}[\mathbf{u}_N^k(\boldsymbol{\mu})])$, $\forall m = 1,\dots,N_{\text{EIM}}$, which implies, as written in Section 5.3.2, RN_{EIM} evaluation of the flux. All the other terms are computed previously and stored: $\alpha_{\text{RB},i}^k(\boldsymbol{\mu})$ are the coefficient of the previous time step, $\boldsymbol{\varphi}_{\text{RB},i}$ are the basis functions of RB, previously computed, and θ_i^m are the projection coefficient of EIM functions onto RB. Overall, the computational cost of a reduced solution at each time step will be $\mathcal{O}(RN_{\text{EIM}})$ flux evaluations and $\mathcal{O}(N_{\text{EIM}}N)$ multiplications.

5.3.6 Error indicator

We can provide an error indicator, which is also an error upper bound for the difference between the high fidelity solution and the reduced one, under some hypothesis. This estimation is derived following the guidelines of [50] and [67]. The hypothesis under which the indicator becomes a bound is that there exists a higher order empirical interpolation of the used operators which is exact. This requirement is fulfilled if we take the interpolation over all the DoFs ($N'_{\text{EIM}} : N_{\text{EIM}} + N'_{\text{EIM}} = H$), where H is the number of DoFs. But, for practical purposes, it has been show in [50] that fewer points are necessary to get a good indicator.
Let us define other N'_{EIM} EIM basis functions $\{\mathbf{q}'_m\}_{m=1}^{N'_{\text{EIM}}}$, simply iterating further the EIM procedure. And we suppose that

$$
\mathcal{I}_{N_{\text{EIM}}+N'_{\text{EIM}}}[\mathcal{L}(\mathbf{x},t^k;\boldsymbol{\mu})][\mathbf{u}_N^k(\boldsymbol{\mu}_i)] = \mathcal{L}(\mathbf{x},t^k;\boldsymbol{\mu})[\mathbf{u}_N^k(\boldsymbol{\mu}_i)]. \tag{5.15}
$$

Moreover, we suppose that the projection of the initial condition are in the reduced basis space, i.e. $\mathbf{u}_h^0(\boldsymbol{\mu}) \in \text{RB}$, $\forall \boldsymbol{\mu} \in \mathcal{P}$. This can be easily obtained if there exists an affine decomposition of the parametric dependent part of the initial conditions: $\mathbf{u}_h^0(\mathbf{x},\boldsymbol{\mu}) = \sum_{k=1}^F \alpha_k(\boldsymbol{\mu})u_k(\mathbf{x})$. Anyway, we will show that, also without fulfilling this condition, the numerical results do not present particular problems if the tolerance of the RB is enough small.
Then, we need a very last hypothesis on the operator $\text{Id} - \Delta t\mathcal{L}(\mathbf{x},t^k;\boldsymbol{\mu})$ namely, to be Lipschitz continuous with constant $C > 0$, i.e. $\forall u, v \in \mathcal{W}_h$:

$$
||u - v - \Delta t\mathcal{L}[u] + \Delta t\mathcal{L}[v]||_{\mathcal{W}_h} \leq C||u - v||_{\mathcal{W}_h} \tag{5.16}
$$

holds.
Under these hypothesis we can say that the error $e^k(\boldsymbol{\mu}) := \mathbf{u}_h^k(\boldsymbol{\mu}) - \mathbf{u}_N^k(\boldsymbol{\mu})$ can be bounded by $\eta^k_{N,N_{\mathrm{EIM}},N'_{\mathrm{EIM}}}(\boldsymbol{\mu})$, which can be computed efficiently, and it is defined as

$$||e^K(\boldsymbol{\mu})||_{\mathcal{W}_h} \leq \eta^K_{N,N_{\mathrm{EIM}},N'_{\mathrm{EIM}}}(\boldsymbol{\mu}) := \sum_{k=1}^{K} C^{K-k} \left(\sum_{m=1}^{N'_{\mathrm{EIM}}} \Delta t \theta_m^k(\boldsymbol{\mu}) \left\| \mathbf{q}'_m \right\|_{\mathcal{W}_h} + \Delta t ||R^k(\boldsymbol{\mu})||_{\mathcal{W}_h} \right), \tag{5.17}$$

where

$$\Delta t R^k(\boldsymbol{\mu}) := \mathbf{u}_N^k(\boldsymbol{\mu}) - \mathbf{u}_N^{k-1}(\boldsymbol{\mu}) + \Delta t \mathcal{I}_{N_{\mathrm{EIM}}}[\mathcal{L}][\mathbf{u}_N^{k-1}(\boldsymbol{\mu})] \tag{5.18}$$

and the coefficient

$$\theta_m^k(\boldsymbol{\mu}) = \boldsymbol{\tau}_m^{N'_{\mathrm{EIM}}} \left(\mathcal{L}[\mathbf{u}_N^{k-1}(\boldsymbol{\mu})] \right), \forall m = 1, \dots, N'_{\mathrm{EIM}}. \tag{5.19}$$

Proof. For the sake of simplicity, we will drop all the $\boldsymbol{\mu}$ parameters.

$$\begin{aligned} \left\| \mathbf{u}_h^{K+1} - \mathbf{u}_N^{K+1} \right\| &= \left\| (\mathrm{Id} - \Delta t \mathcal{L})(\mathbf{u}_h^K) - (\mathrm{Id} - \Delta t \mathcal{I}_{N_{\mathrm{EIM}}}[\mathcal{L}])(\mathbf{u}_N^K) - \Delta t R^K \right\| = \\ &\leq \left\| (\mathrm{Id} - \Delta t \mathcal{L})(\mathbf{u}_h^K) - (\mathrm{Id} - \Delta t \mathcal{L})(\mathbf{u}_N^K) \right\| + \left\| (\Delta t \mathcal{L} - \Delta t \mathcal{I}_{N_{\mathrm{EIM}}}[\mathcal{L}])(\mathbf{u}_N^K) \right\| + \\ &\quad + ||\Delta t R^k||. \end{aligned} \tag{5.20}$$

Then we can use Lipschitz condition (5.16) and get the following:

$$\left\| \mathbf{u}_h^{K+1} - \mathbf{u}_N^{K+1} \right\| \leq C \left\| \mathbf{u}_h^K - \mathbf{u}_N^K \right\| + \left\| (\Delta t \mathcal{L} - \Delta t \mathcal{I}_{N_{\mathrm{EIM}}}[\mathcal{L}])(\mathbf{u}_N^K) \right\| + ||\Delta t R^K||. \tag{5.21}$$

Now, using the fact that the evolution is exactly represented with the second EIM interpolant (5.15), we can rewrite it into:

$$\begin{aligned} C \left\| \mathbf{u}_h^K - \mathbf{u}_N^K \right\| &+ \left\| (\Delta t \mathcal{I}_{N_{\mathrm{EIM}}+N'_{\mathrm{EIM}}}[\mathcal{L}] - \Delta t \mathcal{I}_{N_{\mathrm{EIM}}}[\mathcal{L}])(\mathbf{u}_N^K) \right\| + ||\Delta t R^K|| \leq \\ &\leq C \left\| \mathbf{u}_h^K - \mathbf{u}_N^K \right\| + \left\| \Delta t \sum_{m=1}^{N'_{\mathrm{EIM}}} \boldsymbol{\tau}_m^{N'_{\mathrm{EIM}}} [\mathcal{L}(\mathbf{u}_N^K)] \mathbf{q}'_m \right\| + ||\Delta t R^K|| \leq \\ &\leq C \left\| \mathbf{u}_h^K - \mathbf{u}_N^K \right\| + \left\| \Delta t \sum_{m=1}^{N'_{\mathrm{EIM}}} \theta_m^K \mathbf{q}'_m \right\| + ||\Delta t R^K|| \leq \\ &\leq \sum_{k=1}^{K+1} C^{K+1-k} \left(\left\| \sum_{m=1}^{N'_{\mathrm{EIM}}} \Delta t \theta_m^k(\boldsymbol{\mu}) \mathbf{q}'_m \right\| + ||\Delta t R^k(\boldsymbol{\mu})|| \right). \end{aligned} \tag{5.22}$$

This proves that the error indicator is an actual bound when all the hypothesis are fulfilled. □

Anyway, from experimental results, we can see that, also when we are not in this case, the indicator is giving a good approximation of the error. Indeed, for EIM′, as shown in [50], we can take very few basis functions and get good results, because the chosen DoFs should be the ones that maximize the error. Moreover, its computational cost is $\mathcal{O}(RN'_{\mathrm{EIM}})$ evaluations of the flux.

Estimation of the Lipschitz constant

A couple of words should be spent on the way to find the Lipschitz constant C. Actually, it really depends on the specific method that is used and it is difficult to give a general way to estimate it. For the scheme that we use, we could not find a sharp estimation, because it involves some operators that do not belong to $\mathcal{C}^1$. But, since the operator $\mathcal{L}$ is the discretized operator of the gradient of the flux, we can use the spectral radius ρ of the Jacobian of the flux to approximate this constant.

$$\begin{aligned} &||u-v-\mathcal{L}[u]+\Delta t\mathcal{L}[v]|| \approx ||u-v||+\Delta t||\nabla f(u)-\nabla f(v)|| \approx \\ \approx &||u-v||+\Delta t||J(f)(u-v)|| \leq ||u-v||+\rho\Delta t||u-v|| = (1+\rho\Delta t)||u-v||. \end{aligned} \tag{5.23}$$

What we used in the numerical experiments is a bound b for the spectral radius of the Jacobian of the flux, for u being in a reasonable box. Then we can fix $C = 1 + b\Delta t$. This can be done in a smarter way and more efficiently if the flux is affinely depending on the parameter $\boldsymbol{\mu}$. Therefore, one can split this constant into a parameter dependent and a fixed part.

5.4 Applications to Uncertainty Quantification

5.4.1 Stochastic conservation laws

Many problems in physics and engineering are modeled by hyperbolic systems of conservation or balance laws. As examples for these equations, we can mention the Euler equations of compressible gas dynamics, the Shallow Water Equations of hydrology, the Magnetohydrodynamics (MHD) equations of plasma physics, see, e.g. [43, 56].

Many efficient numerical methods have been developed to approximate the entropy solutions of systems of conservation laws [56, 88], e.g. finite volume or discontinuous Galerkin methods. The classical assumption in designing efficient numerical methods is that all the input data, e.g. initial and boundary conditions, flux vectors, sources, etc, are deterministic. However, in many situations of practical interest, these data are subject to inherent uncertainty in modeling and measurements of physical parameters. Such incomplete information in the uncertain data can be represented mathematically as random fields. Such data are described in terms of statistical quantities of interest like the mean, variance, higher statistical moments; in some cases the distribution law of the stochastic data is also assumed to be known.

A mathematical framework of *random entropy solutions* for scalar conservation laws with random initial data has been developed in [102]. There, existence and uniqueness of random entropy solutions has been shown for scalar hyperbolic conservation laws, also in multiple dimensions. Furthermore, the existence of the statistical quantities of the random entropy solution such as the statistical mean and k-point spatio-temporal correlation functions under suitable assumptions on the random initial data have been proven. The existence and uniqueness of the random entropy solutions for scalar conservation laws with random fluxes has been proven in [101].

A number of numerical methods for uncertainty quantification (UQ) in hyperbolic conservation laws have been proposed and studied recently in e.g. [6, 60, 89, 90, 102, 103, 110, 125, 129, 132, 133].

5.4.2 Random fields and probability spaces

We introduce a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with Ω being the set of all elementary events, or space of outcomes, and $\mathcal{F}$ a σ-algebra of all possible events, equipped with a probability measure $\mathbb{P}$. Random entropy solutions are random functions taking values in a function space; to this end, let $(E, \mathcal{G}, \mathbb{G})$ denote any measurable space. Then an E-valued random variable is any mapping $Y : \Omega \to E$ such that $\forall A \in \mathcal{G}$ the preimage $Y^{-1}(A) = \{\omega \in \Omega : Y(\omega) \in A\} \in \mathcal{F}$, i.e. such that Y is a $\mathcal{G}$-measurable mapping from Ω into E.

We confine ourselves to the case that E is a complete metric space; then $(E, \mathcal{B}(E))$ equipped with a Borel σ-algebra $\mathcal{B}(E)$ is a measurable space. By definition, E-valued random variables $Y : \Omega \to E$ are $\big(E, \mathcal{B}(E)\big)$ measurable. Furthermore, if E is a separable Banach space with norm $\| \circ \|_E$ and with topological dual E^*, then $\mathcal{B}(E)$ is the smallest σ-algebra of subsets of E containing all sets

$$\{x \in E : \varphi(x) < \alpha\}, \varphi \in E^*, \alpha \in \mathbb{R} \, .$$

Hence, if E is a separable Banach space, $Y : \Omega \to E$ is an E-valued random variable if and only if for every $\varphi \in E^*$, $\omega \mapsto \varphi\big(Y(\omega)\big) \in \mathbb{R}$ is an $\mathbb{R}$-valued random variable. Moreover, there hold the following results on existence and uniqueness [102].

For a simple E-valued random variable Y and for any $B \in \mathcal{F}$ we set

$$\int_B Y(\omega)\, \mathbb{P}(d\omega) = \int_B Y \, d\mathbb{P} = \sum_{i=1}^{N} x_i \mathbb{P}(A_i \cap B). \tag{5.24}$$

For such $Y(\omega)$ and all $B \in \mathcal{F}$ holds

$$\Big\| \int_B Y(\omega)\, \mathbb{P}(d\omega) \Big\|_E \leq \int_B \|Y(\omega)\|_E \, \mathbb{P}(d\omega). \tag{5.25}$$

For any random variable $Y : \Omega \to E$ which is Bochner integrable, there exists a sequence $\{Y_m\}_{m \in \mathbb{N}}$ of simple random variables such that, for all $\omega \in \Omega, \|Y(\omega) - Y_m(\omega)\|_E \to 0$ as $m \to \infty$. Therefore (5.24) and (5.25) can be extended to any E-valued random variable. We denote the expectation of Y by

$$\mathbb{E}[Y] = \int_\Omega Y(\omega)\, \mathbb{P}(d\omega) = \lim_{m \to \infty} \int_\Omega Y_m(\omega) \mathbb{P}(d\omega) \in E,$$

and the variance of Y is defined by

$$\mathbb{V}[Y] = \mathbb{E}\big[(Y - \mathbb{E}[Y])^2\big].$$

Denote by $L^p(\Omega, \mathcal{F}, \mathbb{P}; E)$ for $1 \leq p \leq \infty$ the Bochner space of all p-summable, E-valued random variables Y and equip it with the norm

$$\|Y\|_{L^p(\Omega;E)} = \big(\mathbb{E}[\|Y\|_E^p]\big)^{1/p} = \left(\int_\Omega \|Y(\omega)\|_E^p \, \mathbb{P}(d\omega)\right)^{1/p}.$$

For $p = \infty$ we can denote by $L^\infty(\Omega, \mathcal{F}, \mathbb{P}; E)$ the set of all E-valued random variables which are essentially bounded and equip this space with the norm

$$\|Y\|_{L^\infty(\Omega;E)} = \operatorname*{ess\,sup}_{\omega\in\Omega} \|Y(\omega)\|_E.$$

Consider now the balance law (5.1) and assume that the parameter $\boldsymbol{\mu}$ represents vector of real-valued real variables. Different uncertainty quantification (UQ) techniques can be applied to model the effects of this randomness in $\boldsymbol{\mu}$ on the solution $\mathbf{u}$.

5.4.3 Monte Carlo method

In this chapter, we restrict ourselves to the applications of ROM techniques to UQ problems in conjunction with the well-known Monte Carlo sampling method. We note, however, that the outlined ideas could be easily extended to more recent sampling methods such as Multi-Level Monte Carlo (MLMC) method, as well as Stochastic Collocation methods.

The idea of the Monte Carlo method consists in generating M independent, identically distributed samples $\bar{\boldsymbol{\mu}}^i$ of the random variable $\boldsymbol{\mu}$, for $i = 1, \ldots, M$, and calculating the corresponding deterministic approximate solutions $\bar{\mathbf{u}}^i$ of (5.1). Then, the Monte Carlo estimate of the expected solution value $\mathbb{E}[\mathbf{u}]$ at time t and at point x is given by

$$E_M[\mathbf{u}(x,t)] = \frac{1}{M}\sum_{i=1}^{M} \bar{\mathbf{u}}^i(x,t), \tag{5.26}$$

and the variance can be computed according to the unbiased estimate

$$V_M[\mathbf{u}(x,t)] = \frac{1}{M-1}\sum_{i=1}^{M} \left(\bar{\mathbf{u}}^i(x,t) - E_M\right)^2. \tag{5.27}$$

5.5 Numerical results

In this chapter we will present our numerical results that illustrate the behavior of the RB methods in the case of nonlinear unsteady hyperbolic conservation laws in 1D and 2D with applications in UQ.

5.5.1 Stochastic unsteady Burgers' equation in 1D with random data

We consider here Burgers' equations with randomness in both flux and initial data

$$\frac{\partial u}{\partial t} + \frac{\partial f(u,w)}{\partial x} = 0,\ \ x \in [0,\pi],\ \ w \in \Omega, \tag{5.28}$$

$$u_0(x,w) = u_0(x, Y_1(w), Y_2(w)), \tag{5.29}$$

defined on $D = [0, \pi] \subset \mathbb{R},\ t > 0$ with periodic boundary conditions, the nonlinear flux is given as:

$$f(u, w) = f(u, Y_3(w)) = Y_3(w) f(u) = Y_3(w) \frac{u^2}{2} \tag{5.30}$$

and the initial condition is given by:

$$u_0(x, Y_1(w), Y_2(w)) = |\sin(2x + Y_1(w))| + 0.1 Y_2(w), \tag{5.31}$$

where $y_j = Y_j(w),\ j = 1, 2, 3,\ w \in \Omega$ and Y_j is a random variable which takes values in the domain $\mathcal{P} \subset \mathbb{R}^q$ of the parametrized probability space.

The PDE is discretized by an upwind first order finite volume scheme. We used an uniform mesh $\{x_{i-1/2}\}_{i=1}^{N_h+1}$, resulting in a HDM of dimension $N_h = 10^3$, with the CFL condition of 0.318, $K = 159$ time iterations, final time $t^K = 0.159$ and time step of 0.001. In this first example, we will use a finite volume approach, in the RD context, since it can be rewritten in this formulation thanks to [4]. With $x_{i-1/2}$ defining the points of the grid, we define the cells $T_i = [x_{i-1/2}, x_{i+1/2}]$ and we consider constant approximation over each cell u_i. The scheme will then read $u_i^{k+1} = u_i^k - \frac{\Delta t}{\Delta x}\left(f_{i+1/2} - f_{i-1/2}\right)$. We are using the numerical Roe fluxes f defined at the cell interface as:

$$f_{i+1/2} = f(u_L, u_R) = \frac{1}{2}\Big[f(u_L) + f(u_R) - |a(u_L, u_R)|(u_R - u_L)\Big], \tag{5.32}$$

where $u_L = u_i$ and $u_R = u_{i+1}$. The Rankine-Hugoniot velocity is

$$a(u_L, u_R) = \frac{f(u_L) - f(u_R)}{u_L - u_R}.$$

This numerical flux choice has the purpose of linearizing the flux f around the cell interface and then using an upwind flux, which has the role of an entropy fix. For Burgers' equations, the Roe flux including the randomness $Y_3(w)$ writes

$$f(u_L, u_R) = \frac{1}{4} Y_3(w)\Big[u_L^2 + u_R^2 - |u_L^2 + u_R^2|(u_R^2 - u_L^2)\Big]. \tag{5.33}$$

We consider now two cases: the first one which consists only in one randomness in the initial data and the second case which contains randomness in the flux and in the initial condition.

5.5.1.1 Stochastic unsteady Burgers' equation with random initial data

In this case, we consider as deterministic $Y_2(w) = Y_3(w) = 1, \forall w \in \Omega$, while $Y_1(w) \sim \mathcal{U}[0.4, 0.5]$ is the only random variable. In the greedy procedure we sampled the training set using an uniform grid on the parameter domain $D_y = [0.4, 0.5]$. We have not used the PODEIM–Greedy algorithm in this test case (the EIM is performed before the POD–Greedy), because the error of the greedy procedure was naturally decreasing without oscillations. The tolerance set for the EIM procedure was 10^{-6} and for the greedy algorithm was 10^{-4}. What we get from offline phase is an EIM space with 61 functions and a RB space of dimension 12 (see Figure 5.5).

For the online phase, we want to compute some statistical moments with arbitrary probability distributions of the uncertainty, such as the solution mean and the variance, as well as the

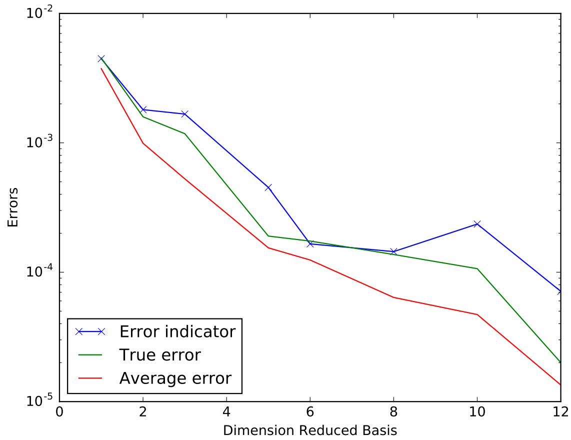


Figure 5.5: The error decrease during basis extension with growing RB size for Burgers' equation with one random data

solution mean plus/minus the standard deviation of the random variable $u_h^K(w)$. This UQ analysis is performed using a set with 100 elements in the parameter domain $D_y = [0.4, 0.5]$, which were generated by a random Monte Carlo method. The advantage of performing an UQ analysis after a RB procedure is that the computational time for a single reduced solution will be much lower than the high fidelity one, the solution accuracy being comparable (see Figure 5.6, 5.7). Indeed, the average computational time for one high fidelity solution is of 1.2551 seconds, while the reduced solution takes only 0.17118 seconds, the percentage of the saved time being then of 86%. [1]

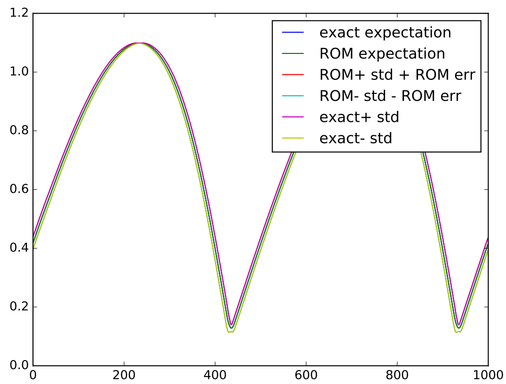


Figure 5.6: Solution mean and the mean plus/minus the standard deviation for both the reduced and the high-fidelity problem in the case of Burgers' equation with one random data

[1]The computations are performed with a Intel(R) Xeon(R) CPU E7-2850 @ 2.00GHz

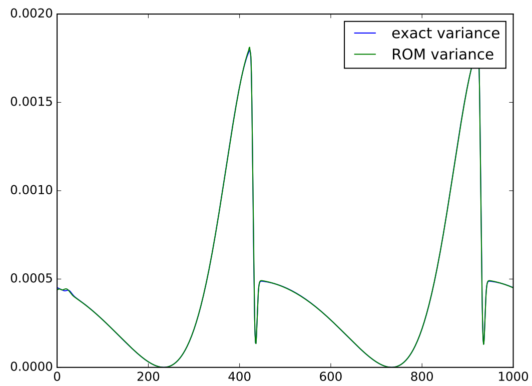


Figure 5.7: Variance for the reduced and the high-fidelity problem in the case of Burgers' equation with one random data

5.5.1.2 Stochastic unsteady Burgers' equation with random flux and initial data

Consider now the case of Burgers' equation with randomness in both flux and initial condition, namely $Y_3(w)$, respectively $Y_1(w)$ and $Y_2(w)$. Let us define $Y_1 \sim \mathcal{U}[0.4, 0.5]$, $Y_2 \sim \mathcal{U}[1, 1.2]$, $Y_3 \sim \mathcal{U}[0.9, 1.1]$. In the greedy procedure we sampled the training set using an uniform three-dimensional grid on the parameter domain $D_y = [0.4, 0.5] \times [1, 1.2] \times [0.9, 1.1]$. We are using the same tolerances for the construction of the EIM space and of the RB as in the previous test case and without using any PODEI algorithm, we obtain an EIM space with 48 functions and an RB space of dimension 11 (see Figure 5.8).

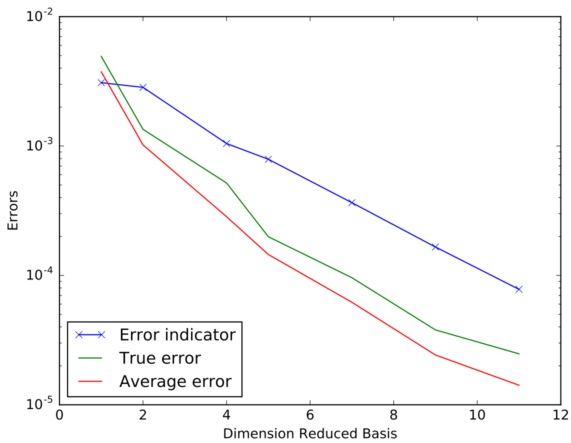


Figure 5.8: The error decrease during basis extension with growing RB size for Burgers' equation with random flux and random initial condition

In the online phase, the UQ analysis is performed using a set with 125 elements in the parameter domain $D_y = [0.4, 0.5] \times [1, 1.2] \times [0.9, 1.1]$, which were generated by a random Monte Carlo method. Comparing again the solution mean and the variance, as well as the solution mean plus/minus the standard deviation of a random variable $u_h^K(w)$ in the case of the reduced problem and the high fidelity one (see Figure 5.9, 5.10), we obtain a computational saving time of 88%. Indeed, the average computational time for one high fidelity solution is of 1.2143 seconds, while the reduced solution takes only 0.14472 seconds.

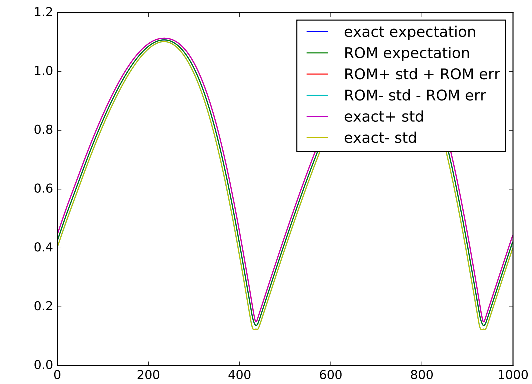


Figure 5.9: Solution mean and the mean plus/minus the standard deviation for both the reduced and the high-fidelity problem in the case of Burgers' equation with random flux and random initial condition

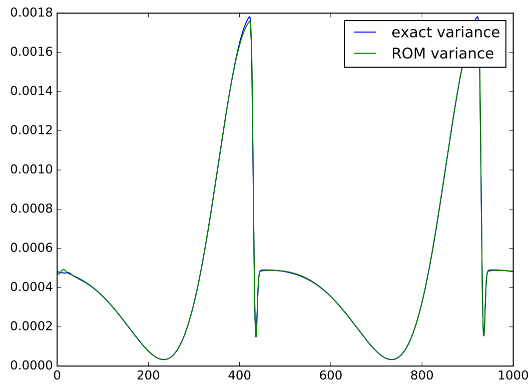


Figure 5.10: Variance for the reduced and the high-fidelity problem in the case of Burgers' equation with random flux and random initial condition

5.5.2 Stochastic Euler equations in 1D with random data

We consider the parametrized Euler equations

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u}, w)}{\partial x} = 0, \ x \in [-1, 1] \tag{5.34}$$

$$\mathbf{u}_0(x, w) = \mathbf{u}_0(x, Y_1(w)) \tag{5.35}$$

with $y_j = Y_j(w), \ j = 1, 2 \ w \in \Omega$ and

$$\mathbf{u} = (\rho, \rho u, E)^T, \ \mathbf{f} = (\rho, \rho u^2 + p, \rho u(E + p))^T, \ p = (\gamma - 1)(E - \frac{1}{2}\rho u^2).$$

We also assume the randomness in the adiabatic constant, $\gamma = Y_2(w)$, and therefore the flux is parameter dependent:

$$\mathbf{f}(\mathbf{u}, w) = \mathbf{f}(\mathbf{u}, Y_2(w)).$$

We consider again two cases: the first one when we have randomness only in the initial data and the second case when we have randomness in the initial data and also in the specific heat ratio γ.

5.5.2.1 Stochastic Euler equations in 1D with random initial data

For this smooth test case, we consider the following random initial condition:

$$\mathbf{u}_0(x, Y_1(w)) = \Big(2+\sin(30Y_1(w))\sin(\pi(x-1)+Y_1(w)), 0, (2+\sin(30Y_1(w))\sin(\pi(x-1)+Y_1(w)))^{\gamma}\Big).$$

We set the value of the specific heat to $\gamma = Y_2(w) = 1.4$ and we construct $Y_1(w)$ using a random Monte Carlo sampling method in the interval $D_y = [0.4, 0.5]$, resulting in a set with 100 elements. The PDE is discretized by a first order finite volume scheme with MUSCL extrapolation on the characteristic variables and minmod limiter on all waves and the resulting HDM is of dimension $N_h = 1200$ using $K = 200$ time iterations of step 0.001, final time $t^K = 0.2$ and the space step of 0.001667.

In the offline step, the tolerance set for the greedy algorithm is $5 \cdot 10^{-6}$ and we are using a PODEIM–Greedy algorithm generating an EIM space with $(10, 11, 10)$ basis and a RB space of dimension $(9, 10, 9)$ in each component, namely in density, momentum and total energy (see Figure 5.11 for the total energy). The PODEIM–Greedy algorithm helps us to avoid the unstable behaviour of the scheme. Indeed, if the accuracy of the empirical interpolation is not enough with respect to the accuracy of the RB space, namely we see an increment in the error, then we discard the newly computed RB functions. This will lead to an automatic control of the correlation between the dimension of the EIM space N_{EIM} and the one of the RB space N, as seen also for this test case.

In the online phase, the UQ analysis is performed using a set with 100 samples in the parameter domain $D_y = [0.4, 0.5]$, which were generated by a random Monte Carlo method. Comparing again the solution mean and the variance, as well as the solution mean plus/minus the standard deviation of a random variable $\mathbf{u}_h^K(w)$ in the case of the reduced problem and the high fidelity one (see Figures 5.12, 5.13, 5.14), we obtain a computational saving time

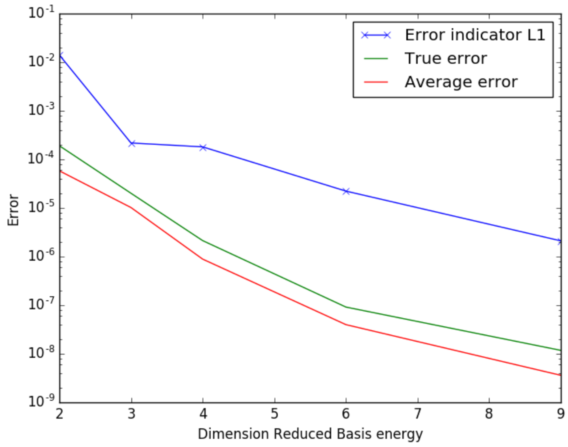
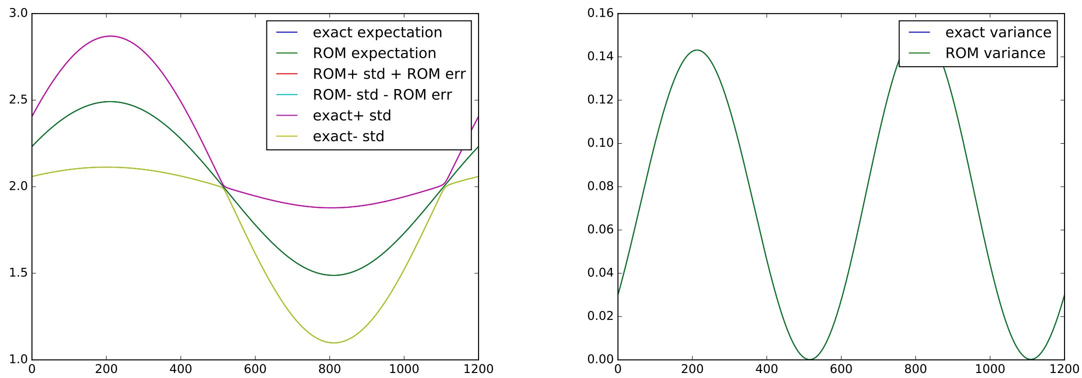


Figure 5.11: The error decrease during basis extension with growing RB size for the total energy component of Euler equation with one random data

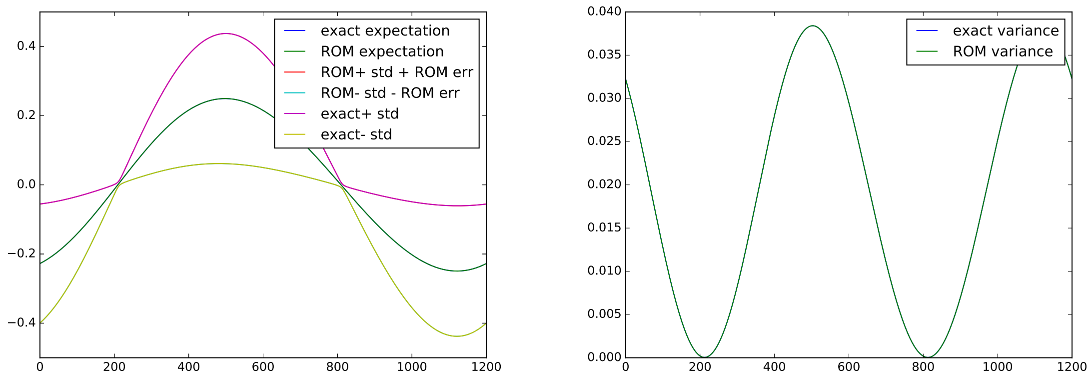
of 89%. For a better visualization, we plot each component of the solution independently. Indeed, the average computational time for one high fidelity solution is of 28.107 seconds, while the reduced solution takes only 3.2133 seconds.



(a)

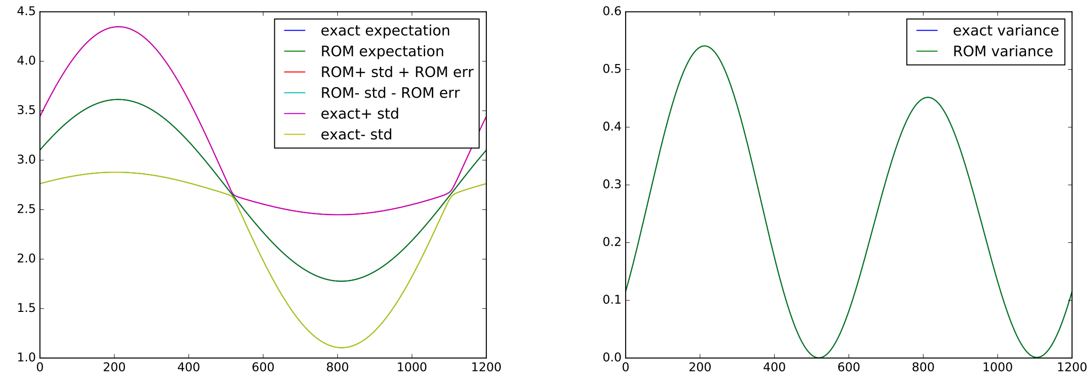
(b)

Figure 5.12: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition for density



(a) (b)

Figure 5.13: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition for momentum



(a) (b)

Figure 5.14: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition for the total energy

5.5.2.2 Stochastic Sod's shock tube problem in 1D with random initial data and random flux

Consider now the Riemann problem for the one-dimensional Euler equations (5.34) with the following initial data set in primitive variables:

$$\mathbf{w}_0(x,w) = (\rho_0(x,w), u_0(x,w), p_0(x,w))^T = \begin{cases} (1,0,1), & \text{if } x < 0 \\ (0.125 + Y_1(w), 0, 0.1), & \text{if } x > 0. \end{cases}$$

In this test case, we have randomness in both flux and initial condition, namely the adiabatic constant $\gamma = Y_2(w)$, respectively $Y_1(w)$. We construct the random variables $Y_1(w), Y_2(w)$ using a random Monte Carlo sampling method in the interval $D_y = [-0.02, 0.02] \times [1.4, 1.5]$, resulting in a set with 100 samples. The PDE is discretized by a first order finite volume scheme with MUSCL extrapolation on the characteristic variables and minmod limiter on all waves and the resulting HDM is of dimension $N_h = 1200$ using $K = 320$ time iterations of step 0.0005, final time $t^K = 0.16$ and the space step of 0.001667.

In the offline step, the tolerance set for the greedy algorithm is $4 \cdot 10^{-6}$ and we are using a PODEI algorithm generating an EIM space with $(68, 83, 89)$ basis and a RB space of dimension $(60, 88, 75)$ in each component, namely in density, momentum and total energy (see Figure 5.15 for the total energy).

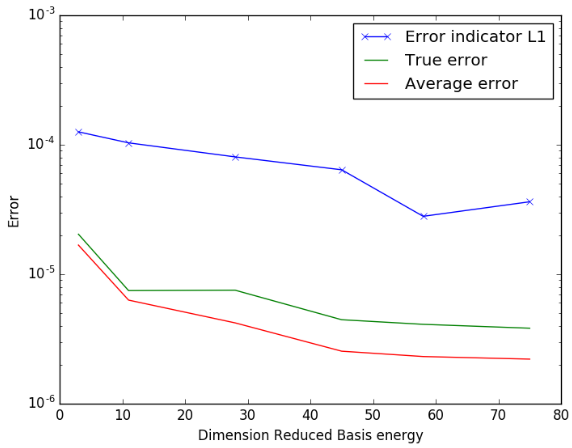


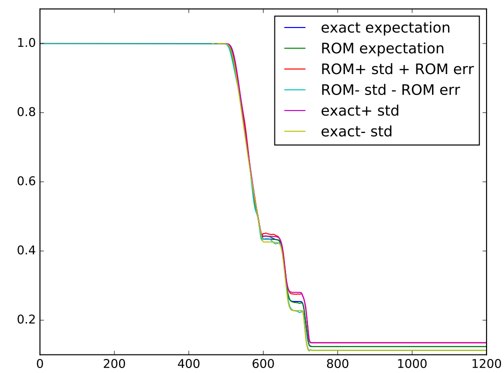
Figure 5.15: The error decrease during basis extension with growing RB size for the total energy component of Euler equation with one random data

In the online phase, the UQ analysis is performed using a set with 100 elements in the parameter domain $D_y = [-0.02, 0.02] \times [1.4, 1.5]$, which were generated by a random Monte Carlo method. Comparing again the solution mean and the variance, as well as the solution mean plus/minus the standard deviation of a random variable $\mathbf{u}_h^K(w)$ in the case of the reduced problem and the high fidelity one (see Figures 5.16, 5.17, 5.18), we obtain a computational saving time of 69%. For a better visualization, we plot each component of the solution independently. Indeed, the average computational time for one high fidelity solution is of 39.448 seconds, while the reduced solution takes only 12.420 seconds.

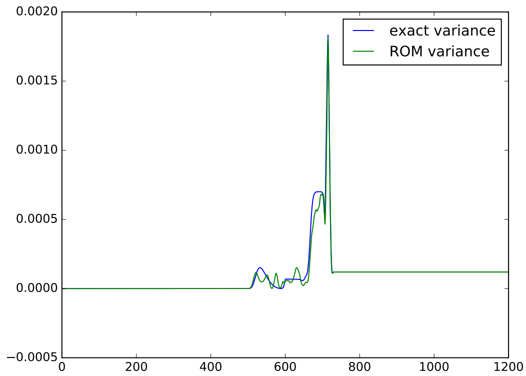
5.5.3 Stochastic Sod's shock tube problem in 2D with random initial data and random flux

Consider the two-dimensional Euler equations with random initial data and random flux:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u}, w)}{\partial x_1} + \frac{\partial \mathbf{g}(\mathbf{u}, w)}{\partial x_2} = 0, \;\; \mathbf{x} = (x_1, x_2) \in D = \{(x_1, x_2) | x_1^2 + x_2^2 \leq 1\} \tag{5.36}$$

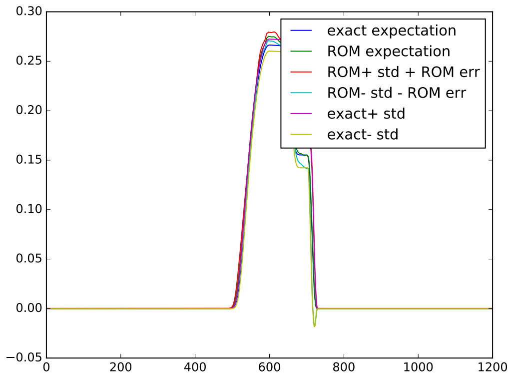


(a)

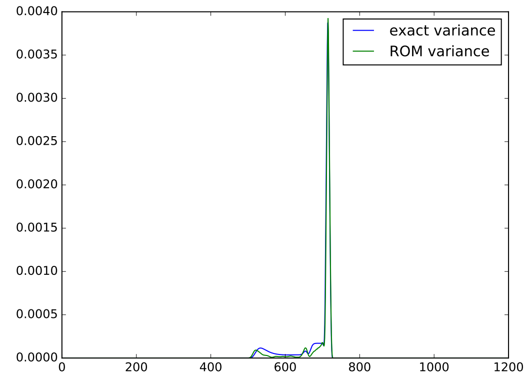


(b)

Figure 5.16: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition and random flux for density



(a)



(b)

Figure 5.17: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition and random flux for momentum

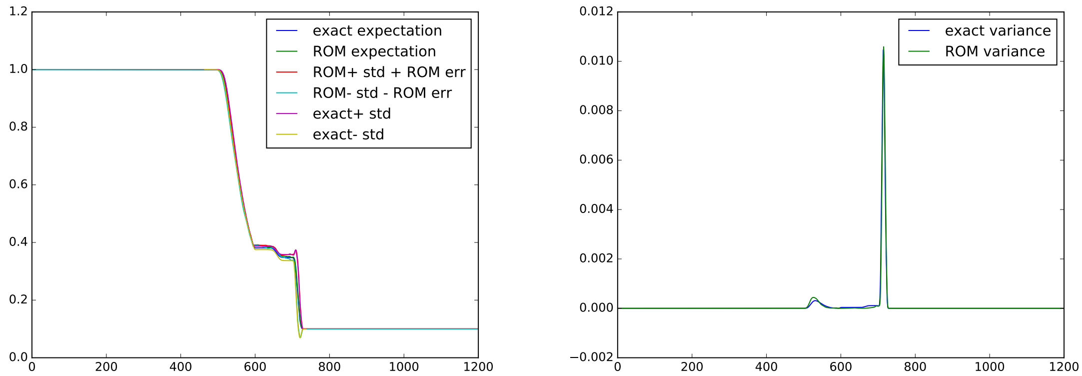
$$\mathbf{u}_0(\mathbf{x}, w) = \mathbf{u}_0(\mathbf{x}, Y_1(w)) \tag{5.37}$$

where $y_j = Y_j(w),\ j = 1, 2,\ w \in \Omega$, the components are expressed as

$$\mathbf{u} = (\rho, \rho u, \rho v, E)^T,\ \mathbf{f} = (\rho, \rho u^2 + p, \rho uv, \rho u(E + p))^T,\ \mathbf{g} = (\rho, \rho uv, \rho v^2 + p, \rho v(E + p))^T$$

and the pressure as

$$p = (\gamma - 1)\Big(E - \frac{1}{2}\rho(u^2 + v^2)\Big).$$



(a) (b)

Figure 5.18: Solution mean, the mean plus/minus the standard deviation and the variance for both the reduced and the high-fidelity problem in the case of Euler equation with random initial condition and random flux for the total energy

We assume again randomness in the adiabatic constant, $\gamma = Y_2(w)$, and therefore

$$\mathbf{f}(\mathbf{u}, w) = \mathbf{f}(\mathbf{u}, Y_2(w))$$

and

$$\mathbf{g}(\mathbf{u}, w) = \mathbf{g}(\mathbf{u}, Y_2(w)).$$

The initial data is set in primitive variables as

$$\mathbf{w}_0(\mathbf{x}, w) = (\rho_0(\mathbf{x}, w), u_0(\mathbf{x}, w), v_0(\mathbf{x}, w), p_0(\mathbf{x}, w))^T = \begin{cases} (1, 0, 0, 1), & \text{if } 0 \leq r < 0.5 \\ (0.125 + Y_1(w), 0, 0, 0.1), & \text{if } 0.5 < r \leq 1 \end{cases}$$

where $r = \sqrt{x_1^2 + x_2^2}$ is the distance of the point (x_1, x_2) from the origin.

The computations have been performed on a triangular mesh consisting of approximately 13000 cells and $N_h = 6775$ DOFs, using $K = 500$ time instances of step 0.0005, the final time is $T = 0.25$ and using a high order RD scheme as presented in [8].

In the offline step, the tolerance set for the greedy algorithm is 0.02 and we are using a PODEIM–Greedy algorithm generating an EIM space with $(67, 68, 69, 76)$ basis functions and a RB space of dimension $(36, 50, 51, 53)$ in each component, namely in density, momentum in x and y direction and total energy. In this test case, we have randomness in both flux and initial condition, namely $Y_2(w)$, respectively $Y_1(w)$. We construct the random variables $Y_1(w), Y_2(w)$ using a random Monte Carlo sampling method in the interval $D_y = [0.125, 0.225] \times [1.4, 1.6]$, resulting in a set with 100 elements. We can see the decay of the error during the Offline phase in Figure 5.19.

In the online phase, the UQ analysis is performed using a set with 50 elements in the parameter domain $D_y = [0.125, 0.225] \times [1.4, 1.6]$, which were generated by a random Monte Carlo method. Comparing again the solution mean (see Figures 5.22, 5.23) and the variance (see

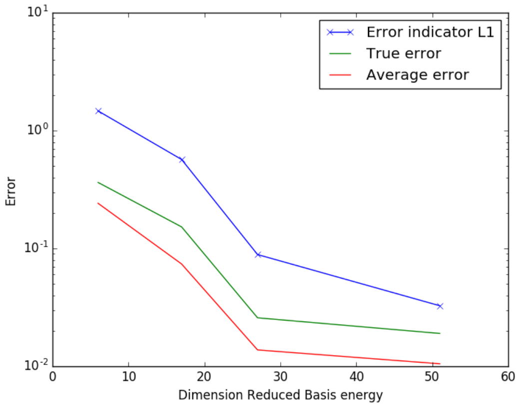
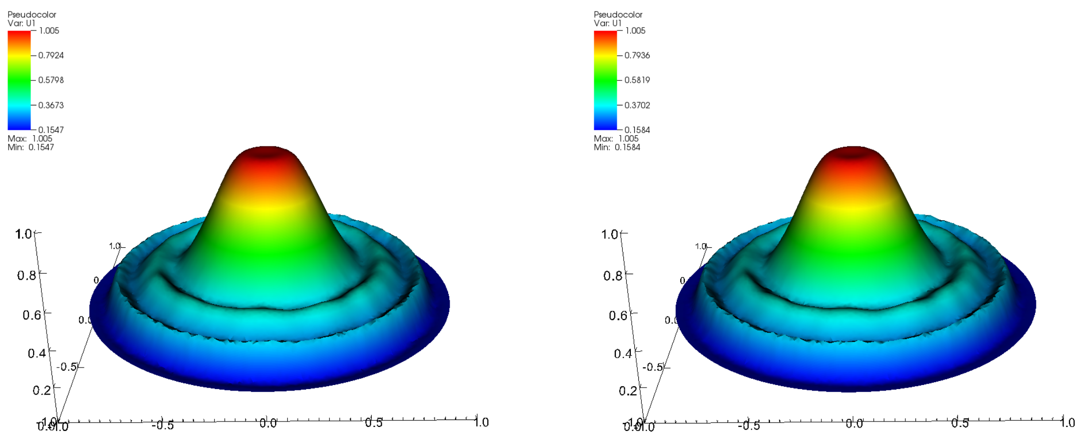


Figure 5.19: Error decay in Offline phase with respect to dimension of reduced basis space of Energy

Figure 5.24, 5.25), in the case of the reduced problem and the high fidelity one (see Figure 5.20, 5.21), we obtain a computational saving time of 76%. Indeed, the average computational time for one high fidelity solution is of 517.59 seconds, while the reduced solution takes only 125.50 seconds.



(a) (b)

Figure 5.20: Density of high-fidelity solution (left) and the reduced solution (right) at final time T=0.25 for $Y = (0.16353811, 1.50632869)$

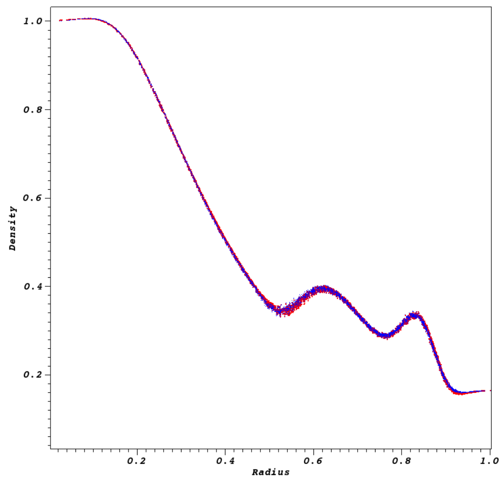
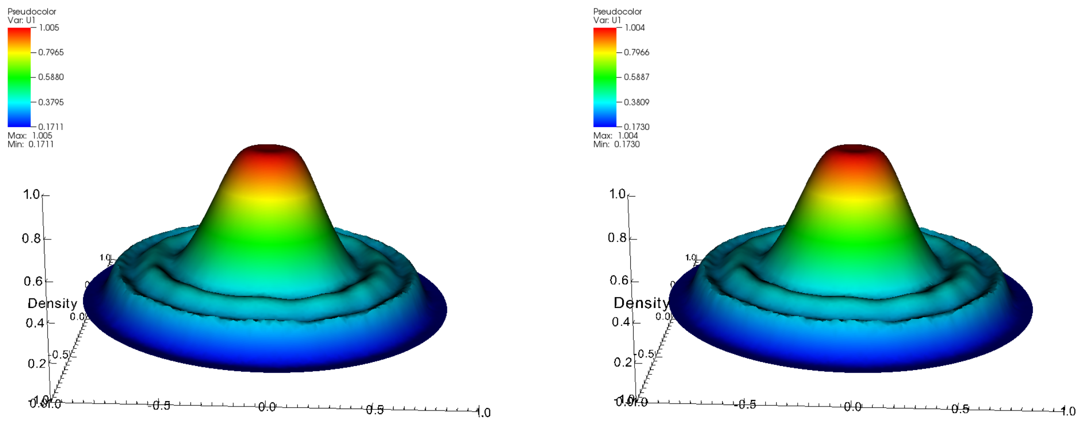


Figure 5.21: Scatter plot of density of the high-fidelity solution (red) and the reduced solution (blue) at final time T=0.25 for $Y = (0.16353811, 1.50632869)$



(a)

(b)

Figure 5.22: Solution mean for density of the high-fidelity problem (left) and for the reduced solution (right) at final time T=0.25

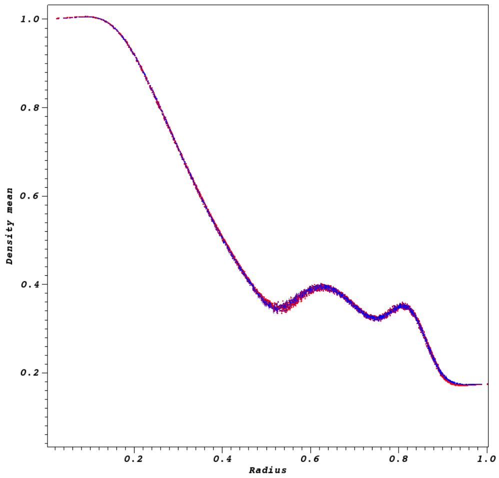
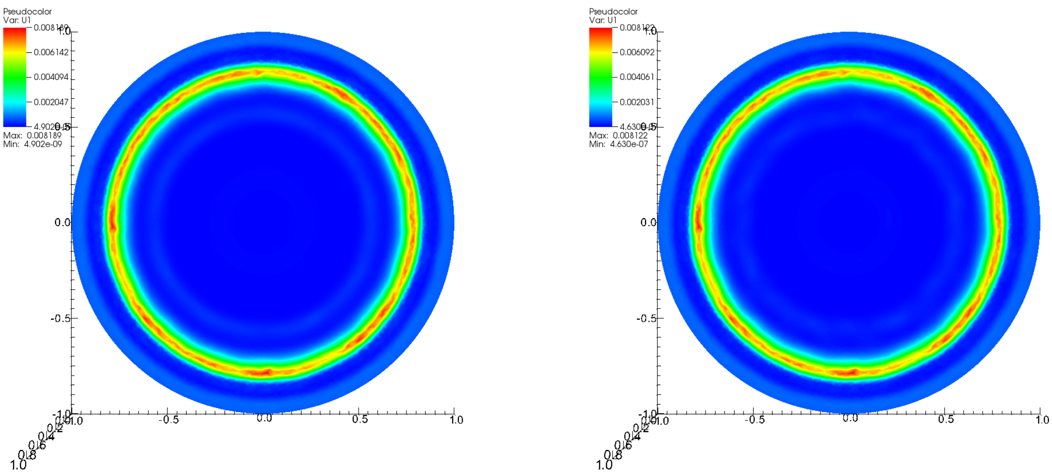


Figure 5.23: Scatter plot of density of the high-fidelity mean solution (red) and the mean of the reduced solution (blue) at final time T=0.25



(a) (b)

Figure 5.24: Variance for the density of high-fidelity problem (left) and for the reduced solution (right) at final time T=0.25

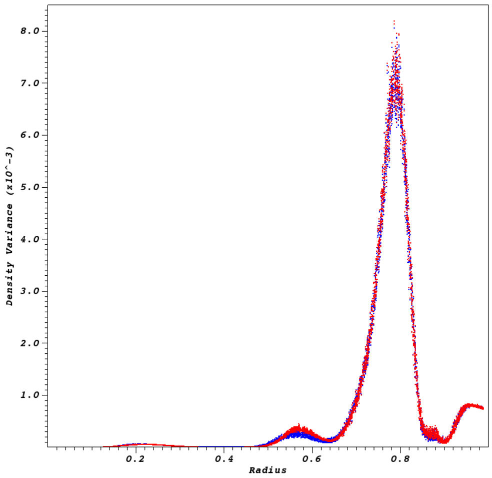


Figure 5.25: Scatter plot of density of the high-fidelity variance (red) and the reduced solution variance (blue) at final time T=0.25

Chapter 6: Conclusions

This chapter summarizes the work conducted in the present dissertation and provides perspectives for future investigation.

6.1 Summary

In this thesis, novel model order reduction techniques for hyperbolic systems of conservation laws were developed. This was a progressive work, where in each phase of research we tackled the problems encountered in the previous step.

Firstly, we have presented a general framework to approximate the solution of steady and unsteady hyperbolic problems using L^1-norm minimization coupled with a dictionary approach. The solution can be smooth or discontinuous, and in the unsteady case, sharp gradients and shocks might exist. Starting from any standard scheme (explicit or implicit), the reduced order solution is obtained at each time step (or each iteration in the implicit case) from a minimization problem in the L^1-norm. We gave a sufficient condition to be able to solve the problem, and we discussed the practical aspects of the method. This method was illustrated by several examples dealing with linear and non linear problems, scalar and systems in one and two space dimensions. A rough error estimate based on the successive projections and the initial solution was given. We also proposed a discussion about hyper-reduction and on the computational cost of the method. In this chapter, when shocks exist in the 2D Euler equations case, the reduced solution presents discrepancies.

Then, in the second chapter, we proposed a complete calibration procedure to make standard RB methods fitted for solving the two dimensional Euler equation around an airfoil. We described an offline calibration procedure and we have shown numerically that it reduces the Kolmogorov N-width and leads to non oscillatory basis. Moreover, this thesis presented also a fully functioning reduced scheme. The computational complexity and the optimization procedures have been theoretically studied and in the end, numerical experiments served as a proof of concept for the global method.

In the last part of this dissertation we focused on MOR methods for parametric nonlinear hyperbolic conservation laws with applications in uncertainty quantification, using all the standard algorithms in RB methods. To generate a RB space, we had to find a low dimensional good approximation of the high fidelity functional space. For this, we used methods as PODEI-Greedy algorithm, by extending the empirical interpolation method basis functions and the POD-Greedy basis functions in a synchronized way.

6.2 Perspectives of Future Work

In this dissertation, a calibration procedure was presented for solving the 2D Euler equations around an airfoil. We have shown numerically that the offline calibration procedure reduces the Kolmogorov N-width but a deeper study of the offline calibration and its effect on the Kolmogorov N-width can be further studied. In the online phase, we have proposed some advanced mappings, where we impose no stretching in the vicinity of the shock. Nevertheless, a numerical investigation can be conducted in order to study the effects of this procedure with respect to the reduced basis. The online procedure described in Section 4.4 it was only theoretical studied. Thus, the construction of a fully reduced scheme could be further investigated, leading to an interesting comparison between L^1-norm and L^2-norm minimization. In Section 4.6.3 we only presented theoretical hyper-reduction ideas which can be applied in this context and we illustrated in Figure 4.9 and in Figure 4.10 one possible output of the greedy algorithm. Hence, the hyper-reduction procedure can be numerically investigated and the conjectures made on the resulting $\hat{\Omega}_{hyper}$, namely that the interesting control volumes are close to the shock, which is fixed in $\hat{\Omega}$ can be further tested. The study of the method on different airfoil shapes is also of a great interest taking in consideration that the Gordon-Hall mapping is a very flexible algorithm. One idea can be to use the NACA 0012 airfoil as reference domain and some other aifoil shapes for the physical domain. This approach could lead to some interesting problems in optimal control.

Another challange that might be further investigated is the ability of reduced basis methods to deal with shock interactions. For multiple shocks, the challange is to deform the geometry, taking in consideration that shocks might interact.

Bibliography

[1] R. Abgrall. Essentially non oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys.*, 214(2):773–808, 2006.

[2] R. Abgrall. Residual distribution schemes: current status and future trends. *Computers & Fluids*, 35(7):641–669, 2006.

[3] R. Abgrall. A review of residual distribution schemes for hyperbolic and parabolic problems: The july 2010 state of the art. *Communications in Computational Physics*, 11(4):1043–1080, 2012.

[4] R. Abgrall. Some remarks about conservation for residual distribution schemes. working paper or preprint, Sept. 2017.

[5] R. Abgrall, D. Amsallem, and R. Crisovan. Robust model reduction by L^1-norm minimization and approximation via dictionnaries: application to non linear hyperbolic problems. *Adv. Model. and Simul. in Eng. Sci.*, 3(1), 2016.

[6] R. Abgrall and P. Congedo. A semi-intrusive deterministic approach to uncertainty quantification in non-linear fluid flow problems. *Journal of Computational Physics*, 235:828 – 845, 2013.

[7] R. Abgrall and D. De Santis. High order residual distribution scheme for Navier-Stokes equations. *AIAA Paper 2011-3231, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, 27-30 June 2011*, pages 1–24, 2011.

[8] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *Journal of Computational Physics*, 230(11):4103 – 4136, 2011. Special issue High Order Methods for CFD Problems.

[9] R. Abgrall and S. Mishra. Uncertainty quantification for hyperbolic systems of conservation laws. Technical Report 2016-58, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2016.

[10] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. *Lecture Notes in Pure and Applied Mathematics*, pages 317–332, 2001.

[11] N. Allahverdi, A. Pozo, and E. Zuazua. Numerical aspects of large-time optimal control of Burgers equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(5):1371–1401, 2016.

[12] D. Amsallem, J. Cortial, and C. Farhat. Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information. *AIAA Journal*, 48(9):2029–2037, 2010.

[13] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat. Model predictive control under coupled fluid-structure constraints using a database of reduced-order models on a tablet. *AIAA Paper 2013-2588, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 26-29, 2013*, pages 1–12, 2013.

[14] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[15] D. Amsallem, M. Zahr, Y. Choi, and C. Farhat. Design optimization using hyper-reduced-order models. *Structural and Multidisciplinary Optimization*, 51(4):919–940, 2015.

[16] D. Amsallem, M. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[17] J. Ausseur, J. Pinier, M. Glauser, and H. Higuchi. Predicting the Dynamics of the Flow over a NACA 4412 using POD. In *APS Division of Fluid Dynamics Meeting Abstracts*, page DN.008, Nov. 2004.

[18] M. Balajewicz, D. Amsallem, and C. Farhat. Projection-based model reduction for contact problems. *International Journal for Numerical Methods in Engineering*, 106:644–663, 2016.

[19] C. Bardos and O. Pironneau. Derivatives and control in the presence of shocks. *Computational Fluid Dynamics Journal*, 11(4):383–391, 2003.

[20] M. F. Barone, I. Kalashnikova, D. Segalman, and H. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *Journal of Computational Physics*, 228(6):1932–1946, 2009.

[21] M. Barrault, Y. Maday, N. Nguyen, and A. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus de l'Academie des Sciences Paris*, 339:667–672, 2004.

[22] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox. *Model Reduction and Approximation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[23] P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban. *Model Reduction of Parametrized Systems*. Springer International Publishing, 2017.

[24] W. Beyn and V. Thümmler. Freezing solutions of equivariant evolution equations. *SIAM Journal on Applied Dynamical Systems*, 3(2):85–116, 2004.

[25] S. Bianchini and A. Bressan. Vanishing viscosity solutions to nonlinear hyperbolic systems. *Ann. of Math*, 161:223–342, 2005.

[26] H. Bijl, D. Lucor, S. Mishra, and C. Schwab. *Uncertainty quantification in computational fluid dynamics*, volume 92. Springer Science & Business Media, 2013.

[27] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM Journal on Mathematical Analysis*, 43(3):1457–1472, 2011.

[28] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[29] A. Bressan. *Hyperbolic systems of conservation laws; The one-dimensional Cauchy problem.* Oxford University Press, Jan. 2000.

[30] S. Brunton, J. Tu, I. Bright, and J. Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.

[31] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 8 2004.

[32] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications. *AIAA Journal*, 46:2520–2529, Oct. 2008.

[33] J. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:174–199, 1948.

[34] N. Cagniart, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects, Nov. 2016.

[35] E. Candes and J. Romberg. Robust signal recovery from incomplete observations. In *Image Processing, 2006 IEEE International Conference on*, pages 1281–1284. IEEE, 2006.

[36] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[37] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, June 2013.

[38] G. Chen. The method of quasi-decoupling for discontinuous solutions of conservation laws. *Arch. Rat. Mech. Anal.*, 121:131–185, 1992.

[39] P. Chen, A. Quarteroni, and G. Rozza. A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 51(6):3163–3185, 2013.

[40] P. Chen, A. Quarteroni, and G. Rozza. Comparison between reduced basis and stochastic collocation methods for elliptic problems. *Journal of Scientific Computing*, 59(1):187–216, Apr 2014.

[41] P. Chen, A. Quarteroni, and G. Rozza. Reduced basis methods for uncertainty quantification. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):813–869, 2017.

[42] P. Colella, M. Dorr, J. Hittinger, and D. Martin. High-order, finite-volume methods in mapped coordinates. *Journal of Computational Physics*, 230(8):2952–2976, 2011.

[43] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Fundamental Principles of Mathematical Sciences*. Springer-Verlag, 2010.

[44] W. Dahmen, C. Plesken, and G. Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):623–663, Jan. 2014.

[45] I. Daubechies, R. DeVore, M. Fornasier, and C. Gunturk. Iteratively re-weighted least squares minimization for sparse recovery. *Communications on pure and applied mathematics*, 63:1–38, 2010.

[46] H. Deconinck and M. Ricchiuto. Residual distribution schemes: foundations and analysis. *Encyclopedia of computational mechanics*, 2007.

[47] R. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in banach spaces. *Constructive Approximation*, 37(3):455–466, 2013.

[48] M. Dihlmann, M. Drohmann, and B. Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. *Proc. of ADMOS*, 2011, 2011.

[49] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[50] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969, 2012.

[51] J. Eftang, M. Grepl, and A. Patera. A posteriori error bounds for the empirical interpolation method. *Comptes Rendus Mathematique*, 348(9):575 – 579, 2010.

[52] M. Fogleman, J. Lumley, D. Rempfer, and D. Haworth. Application of the proper orthogonal decomposition to datasets of internal combustion engine flows. *Journal of Turbulence*, 5:N23, 2004.

[53] J. Gerbeau and D. Lombardi. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *Journal of Computational Physics*, 265:246–269, 2014.

[54] R. Ghanem, D. Higdon, and H. Owhadi. *Handbook of uncertainty quantification.* Springer International Publishing, 2016.

[55] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Communications on Pure and Applied Mathematics*, 18(4):697–715, 1965.

[56] E. Godlewski and P. Raviart. *Hyperbolic systems of conservation laws.* Ellipses, Feb. 1991.

[57] E. Godlewski and P. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Springer, 2014.

[58] S. Godunov. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Math. Sbornik*, 47:271–306, 1959.

[59] W. Gordon and C. Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21(2):109–129, 1973.

[60] D. Gottlieb and D. Xiu. Galerkin method for wave equations with uncertain coefficients. *Commun. Comput. Phys.*, 3:505–518, 2008.

[61] M. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: M2AN*, 41(3):575–605, 2007.

[62] M. Grepl and A. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[63] J. Guermond, F. Marpeau, and B. Popov. A fast algorithm for solving first-order PDEs by L1-minimization. *Communications in Mathematical Sciences*, 6(1):199–216, 2008.

[64] J. Guermond and B. Popov. L^1-Approximation of Stationary Hamilton–Jacobi Equations. *SIAM Journal on Numerical Analysis*, 47(1):339–362, Jan. 2009.

[65] B. Haasdonk and M. Ohlberger. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In *Proc. 5th International Symposium on Finite Volumes for Complex Applications*, pages 471–478.

[66] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.

[67] B. Haasdonk and M. Ohlberger. Reduced basis method for explicit finite volume approximations of nonlinear conservation laws. In *Hyperbolic problems: theory, numerics and applications*, volume 67, pages 605–614. Amer. Math. Soc., 2009.

[68] B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *ETNA, Electronic Transactions on Numerical Analysis*, 32:145–168, 2008.

[69] J. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.

[70] S. Hovland, J. Gravdahl, and K. Willcox. Explicit model predictive control for large-scale systems via model reduction. *Journal of Guidance, Control, and Dynamics*, 31(4):1–23, 2008.

[71] P. Huber and E. Ronchetti. *Robust Statistics*. John Wiley & Sons, Sept. 2011.

[72] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2):022923, 2014.

[73] K. Ito and S. Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, June 1998.

[74] C. Jäggli, L. Iapichino, and G. Rozza. An improvement on geometrical parameterizations by transfinite maps. *Comptes Rendus Mathématique*, 352(3):263–268, 2014.

[75] I. Jolliffe. *Principal Component Analysis*. Springer New York, 2002.

[76] M. Kahlbacher and S. Volkwein. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discussiones Mathematicae, Differential Inclusions, Control and Optimization*, 27(1):95–117, 2007.

[77] I. Kalashnikova and M. Barone. Stable and efficient Galerkin reduced order models for non-linear fluid flow. *AIAA Journal*, 2011.

[78] S. Kaulmann and B. Haasdonk. Online greedy reduced basis construction using dictionaries. *University of Stuttgart*, 2013.

[79] A. Kolmogorov. Über die beste annäherung von funktionen einen gegebenen funktionenklasse. *Ann. Math.*, 37:107–110, 1936.

[80] S. Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217–243, 1970.

[81] K. Kunisch. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 2003.

[82] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[83] J. Lavery. Nonoscillatory solution of the steady-state inviscid Burgers' equation by mathematical programming. *J. Comput. Phys.*, 79(2):436–448, 1988.

[84] P. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, 1954.

[85] P. Lax. Hyperbolic systems of conservation laws II. *Communications on Pure and Applied Mathematics*, 10(4):537–566, 1957.

[86] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[87] P. LeGresley and J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. *AIAA Paper 2000-2545 Fluids 2000 Conference and Exhibit, Denver, CO*, pages 1–14, 2000.

[88] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[89] G. Lin, C.-H. Su, and G. E. Karniadakis. Predicting shock dynamics in the presence of uncertainties. *J. Comput. Phys.*, 217:260–276, 2006.

[90] G. Lin, C.-H. Su, and G. E. Karniadakis. Stochastic modelling of random roughness in shock scattering problems: theory and simulations. *Comp. Meth. App. Mech. Eng.*, 197, 2008.

[91] A. Løvgren, Y. Maday, and E. Rønquist. *The Reduced Basis Element Method for Fluid Flows*, 2006.

[92] A. Løvgren, Y. Maday, and E. Rønquist. Global C1 maps on general domains. *Mathematical Models and Methods in Applied Sciences*, 19(5):803–832, 2009.

[93] A. Løvgren, Y. Maday, and E. Rønquist. The reduced basis element method: Offline-online decomposition in the nonconforming, nonaffine case. In *Spectral and High Order Methods for Partial Differential Equations*, pages 247–254. Springer, 2011.

[94] J. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, 1967.

[95] H. Ly and H. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1-3):223–236, 2001.

[96] Y. Maday, A. Manzoni, and A. Quarteroni. An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems. *Comptes Rendus Mathématique*, 354(12):1188–1194, 2016.

[97] Y. Maday, N. Nguyen, A. Patera, and S. Pau. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8(1):383–404, 2009.

[98] Y. Maday and E. Rønquist. A reduced-basis element method. *Journal of scientific computing*, 17(1-4):447–459, 2002.

[99] Y. Maday and B. Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM Journal on Scientific Computing*, 35(6):A2417–A2441, 2013.

[100] J. Melenk. On n-widths for elliptic problems. *Journal of Mathematical Analysis and Applications*, 247(1):272 – 289, 2000.

[101] S. Mishra, N. Risebro, C. Schwab, and S. Tokareva. Numerical solution of scalar conservation laws with random flux functions. *SIAM/ASA J. Uncertain. Quantif.*, 4:552–591, 2016.

[102] S. Mishra and C. Schwab. Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random intitial data. *Math. Comp.*, 81:1979–2018, 2012.

[103] S. Mishra, C. Schwab, and J. Šukys. Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *J. Comput. Phys.*, 231:3365–3388, 2012.

[104] J. Nečas. *Direct Methods in the Theory of Elliptic Equations*. Springer Verlag, 2012.

[105] J. Nocedal and S. Wright. *Numerical optimization*. Springer, Dec. 2006.

[106] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathématique*, 351(23):901–906, 2013.

[107] O. Oleinik. Discontinuous solutions of nonlinear differential equations. *Usp. Mat. Nauk.*, 12:3–73, 1957. English transl. in AMS Transl. **26:**1155-1163, 1963.

[108] A. Patera and G. Rozza. *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*. MIT-Pappalardo Graduate Monographs in Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, 2007.

[109] W. Philip. n-widths in approximation theory (allan pinkus). *SIAM Review*, 28(2):283–284, 1986.

[110] G. Poëtte, B. Després, and D. Lucor. Uncertainty quantification for systems of conservation laws. *J. Comput. Phys.*, 228:2443–2467, 2009.

[111] C. Prud'Homme, D. Rovas, K. Veroy, L. Machiels, Y. Maday, A. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, Nov. 2001.

[112] C. Prud'homme, D. Rovas, K. Veroy, and A. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM: M2AN*, 36(5):747–771, 2002.

[113] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*. Springer International Publishing, 2016.

[114] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier–Stokes equations by reduced basis methods. *Numerical Methods for Partial Differential Equations*, 23(4):923–948, 2007.

[115] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[116] M. Ricchiuto. An explicit residual based approach for shallow water flows. *Journal of Computational Physics*, 280(Supplement C):306 – 344, 2015.

[117] D. Rim, S. Moe, and R. LeVeque. Transport reversal for model reduction of hyperbolic partial differential equations, Jan. 2017.

[118] C. W. Rowley, T. Colonius, and R. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115 – 129, 2004.

[119] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust. Reduction and reconstruction for self-similar dynamical systems. *Nonlinearity*, 16(4):1257, 2003.

[120] J. E. Rowley, Clarence W.and Marsden. Reconstruction equations and the karhunen–loéve expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1):1 – 19, 2000.

[121] G. Rozza. Shape design by optimal flow control and reduced basis techniques applications to bypass configurations in haemodynamics. page 290, 2005.

[122] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15:229–275, 2008.

[123] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1):75–89, 2009.

[124] R. Schmit, M. Glauser, and S. Gorton. Low dimensional tools for flow-structure interaction problems: Application to micro air vehicles. In *41st Aerospace Sciences Meeting and Exhibit*, 2003.

[125] C. Schwab and S. Tokareva. High order approximation of probabilistic shock profiles in hyperbolic conservation laws with uncertain initial data. *ESAIM: M2AN*, 47:807–835, 2013.

[126] K. Siddiqi, B. Kimia, and C. Shu. Geometric shock-capturing ENO schemes for subpixel interpolation, computation, and curve evolution. In *Computer Vision, 1995. Proceedings., International Symposium on Computer Vision - ISCV*, pages 437–442. IEEE, 1995.

[127] L. Sirovich. Turbulence and the dynamics of coherent structures. Part I: coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[128] T. Taddei, S. Perotto, and A. Quarteroni. Reduced basis techniques for nonlinear conservation laws. *ESAIM: M2AN*, 49(3):787–814, 2015.

[129] S. Tokareva, C. Schwab, and S. Mishra. High order SFV and mixed SDG/FV methods for the uncertainty quantification in multidimensional conservation laws. In R. Abgrall, H. Beaugendre, P. Congedo, C. Dobrzynski, V. Perrier, and M. Ricchiuto, editors, *High order nonlinear numerical schemes for evolutionary PDEs*, volume 99 of *Lecture notes in computational sciences and engineering*. Springer, 2014.

[130] T. Tonn, K. Urban, and S. Volkwein. Optimal control of parameter-dependent convection-diffusion problems around rigid bodies. *SIAM Journal on Scientific Computing*, 32(3):1237–1260, 2010.

[131] E. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer, Berlin, Heidelberg, 1997.

[132] J. Troyen, O. L. M. tre, M. Ndjinga, and A. Ern. Intrusive Galerkin methods with upwinding for uncertain nonlinear hyperbolic systems. *J. Comput. Phys.*, 229:6485–6511, 2010.

[133] J. Troyen, O. L. M. tre, M. Ndjinga, and A. Ern. Roe solver with entropy corrector for uncertain hyperbolic systems. *J. Comput. Phys.*, 235:491–506, 2010.

[134] K. Veroy and A. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigurous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.

[135] K. Veroy, C. Prud'homme, and A. T. Patera. Reduced-basis approximation of the viscous burgers equation: rigorous a posteriori error bounds. *Comptes Rendus Mathematique*, 337(9):619 – 624, 2003.

[136] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. *AIAA Paper 2003-3847, 16th Computational Fluid Dynamics Conference, Orlando Florida, 24-25 June 2003*, 3847, 2003.

[137] K. Washabaugh, D. Amsallem, M. Zahr, and C. Farhat. Nonlinear Model Reduction for CFD Problems Using Local Reduced-Order Bases. *42nd AIAA Fluid Dynamics Conference and Exhibit. New Orleans, Louisiana*, 2012.

[138] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250, 2017.

[139] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.

[140] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA 2001-2611*, pages 1–9, Apr. 2001.